

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E
AUTOMAÇÃO INDUSTRIAL

Regina Mitsue Azuma

**Otimização multiobjetivo em problema de
estoque e roteamento gerenciados pelo fornecedor**

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Orientador: Fernando José Von Zuben

Campinas, SP
2011

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Az85o Azuma, Regina Mitsue
Otimização multiobjetivo em problema de estoque e roteamento gerenciados pelo fornecedor / Regina Mitsue Azuma. --Campinas, SP: [s.n.], 2011.

Orientador: Fernando José Von Zuben.
Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Problema de roteamento de veículos. 2. Otimização combinatória. 3. Controle de estoque. 4. Algoritmos evolutivos. I. Von Zuben, Fernando José. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Evolutionary multi-objective optimization for the vendor-managed inventory routing problem

Palavras-chave em Inglês: Vehicle routing problem, Combinatorial optimization, Inventory control, Evolutionary algorithm

Área de concentração: Engenharia de Computação

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Ricardo Hiroshi Caldeira Takahashi, Leonardo Tomazeli Duarte

Data da defesa: 27/01/2011

Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidata: Regina Mitsue Azuma

Data da Defesa: 27 de janeiro de 2011

Título da Tese: "Otimização multiobjetivo em problema de estoque e roteamento gerenciados pelo fornecedor"

Prof. Dr. Fernando José Von Zuben (Presidente): Fernando José Von Zuben

Prof. Dr. Ricardo Hiroshi Caldeira Takahashi: Ricardo

Prof. Dr. Leonardo Tomazeli Duarte: Leonardo Tomazeli Duarte

Resumo

A classe de problemas de estoque e roteamento está presente em várias áreas, incluindo indústria automobilística e gerência de numerário no reabastecimento de caixas eletrônicos. Supondo que o fornecedor é responsável pela estocagem e distribuição dos produtos, sujeito a um conjunto de restrições, o desafio que se apresenta é a determinação de uma política ótima, mais especificamente quais clientes atender, qual quantidade a ser fornecida a cada cliente e qual rota empregar visando a minimização dos custos. Este trabalho apresenta uma proposta de solução para uma das mais comuns formulações do problema: um produto é distribuído a partir de um fornecedor para vários clientes em um horizonte de tempo definido. O transporte é realizado por um veículo de capacidade limitada. Para produzir a otimização simultânea de ambos os objetivos, minimização dos custos de transporte e estoque, a proposta segue uma abordagem multiobjetivo e se baseia no uso do algoritmo SPEA2 (do inglês, *Strength Pareto Evolutionary Algorithm 2*), incluindo inovações na representação de soluções-candidatas, nos operadores genéticos e de busca local. A fronteira de Pareto estimada é então composta de múltiplas soluções não-dominadas, representando compromissos distintos entre custos de transporte e estoque. Como casos de estudo, são tomadas instâncias de médio porte extraídas da literatura e são geradas instâncias de grande porte. Para as instâncias de médio porte, as fronteiras de Pareto estimadas em cada caso são comparadas com as respectivas soluções ótimas da versão mono-objetivo de cada problema, pois já existe um algoritmo exato de solução para a formulação mono-objetivo de instâncias de médio porte.

Palavras-chave: problema de estoque e roteamento gerenciados pelo fornecedor, otimização multiobjetivo, algoritmos evolutivos.

Abstract

The class of inventory routing problems (IRP) is present in several areas, including automotive industry and cash management for ATM networks. Given that the supplier is responsible for managing the product inventory and replenishment, subject to a set of restrictions, the challenge here is to determine an optimal policy, more specifically which retailers to serve, the quantity to deliver to each retailer and which routes to employ in order to minimize the cost. This work presents a proposal to solve one version of the IRP usually found in the scientific literature: a product is distributed from a supplier to several retailers in a defined time horizon. Shipment is performed by a vehicle with limited capacity. To perform the simultaneous optimization of both objectives, minimization of transportation and inventory costs, the proposal follows a multi-objective approach based on SPEA2 (*Strength Pareto Evolutionary Algorithm 2*), including innovative aspects mainly associated with the representation of candidate solutions, genetic operators and local search. The Pareto front is then composed of multiple non-dominated solutions with distinct trade-offs between transportation and inventory costs. As case studies, medium size instances extracted from the literature are considered and large size instances are generated. For the medium size instances, the estimated Pareto fronts are compared, in each case, with the corresponding optimal solutions associated with the single-objective version of each problem, given that there is already an exact algorithm to solve such medium size single-objective instances.

Keywords: vendor-managed inventory routing problem, multi-objective optimization, evolutionary algorithms.

Agradecimentos

Primeiramente, agradeço ao meu orientador, Prof. Fernando José Von Zuben, pela oportunidade e pela competência com que sempre conduziu esta pesquisa.

Ao colega Guilherme Palermo Coelho, pelos esclarecimentos e contribuições, particularmente no tocante aos conceitos de otimização multiobjetivo.

Aos demais colegas de pós-graduação, pelas críticas e sugestões.

À minha família e ao meu namorado, pelo apoio e incentivo durante esta jornada.

À Motorola, pelas horas cedidas de meu trabalho ao desenvolvimento desta pesquisa.

Ao meu gerente, José Barletta, pela compreensão e confiança que sempre depositou em mim.

Aos meus pais

Sumário

Lista de Figuras	xvii
Lista de Tabelas	xix
Lista de Algoritmos	xxi
1 Introdução e Motivação	1
1.1 Introdução	1
1.2 Motivação da pesquisa	3
1.3 Contribuições da pesquisa	5
1.4 Organização do texto	6
1.5 Síntese do capítulo	6
2 Problema de estoque e roteamento gerenciados pelo fornecedor	7
2.1 Síntese do capítulo	16
3 Otimização multiobjetivo	17
3.1 Introdução	17
3.2 Definição	18
3.3 Metas da otimização multiobjetivo	19
3.4 Abordagens clássicas	20
3.4.1 Soma ponderada dos objetivos	20
3.4.2 Método de restrições ϵ	22
3.4.3 Programação por metas	23
3.4.4 Análise das abordagens tradicionais	23
3.5 Algoritmos evolutivos para otimização multiobjetivo	24
3.6 Métricas de desempenho	26
3.7 SPEA2	29

3.7.1	Função de <i>fitness</i>	30
3.7.2	Algoritmo de corte	31
3.8	Síntese do capítulo	32
4	Caracterização do problema	33
4.1	Descrição	33
4.2	Classificação	37
4.3	Formulação geral	39
4.4	Formulação mono-objetivo \times multiobjetivo	41
4.5	Política de reabastecimento	42
4.6	Instâncias de teste	42
4.6.1	Instâncias de <i>benchmark</i> da literatura	43
4.6.2	Proposta de novas instâncias	45
4.7	Síntese do capítulo	46
5	Estratégia de solução	47
5.1	Critério de dominância	47
5.2	Codificação	50
5.3	Operadores de recombinação	51
5.3.1	Recombinação vertical de um ponto	54
5.3.2	Recombinação horizontal de um ponto	55
5.3.3	Recombinação uniforme de células	56
5.3.4	Recombinação uniforme de colunas	57
5.3.5	Recombinação uniforme de linhas	58
5.4	Operadores de mutação	59
5.4.1	União de reabastecimentos	60
5.4.2	Inserção de reabastecimento	61
5.4.3	Troca de reabastecimentos	62
5.5	Operadores de busca local	62
5.5.1	Remoção de reabastecimentos	64
5.5.2	Deslocamento de reabastecimento	65
5.5.3	Troca de reabastecimento em rota de alto custo	66
5.5.4	Alocação máxima em cliente de baixo custo de estoque	67
5.5.5	União e separação de rotas I	68
5.5.6	União e separação de rotas II	70
5.6	Implementação	71

5.7	Síntese do capítulo	71
6	Resultados	73
6.1	Apresentação de uma solução do conjunto de Pareto	74
6.2	Resultados para as instâncias propostas na literatura	76
6.3	Amostragem de soluções	86
6.4	Resultados para as instâncias de grande porte	89
6.5	Síntese do capítulo	91
7	Conclusão e perspectivas	93
7.1	Considerações finais	93
7.2	Perspectivas e pesquisas futuras	95
	Referências bibliográficas	96

Lista de Figuras

3.1	Exemplo de mapeamento de uma solução no espaço de variáveis, tridimensional, para o espaço de objetivos, bidimensional	18
3.2	O ponto B, assim como qualquer ponto na região hachurada (desde que corresponda a uma solução factível) domina a solução representada pelo ponto A	19
3.3	Metas da otimização multiobjetivo	20
3.4	Método de soma ponderada dos objetivos	21
3.5	Método de restrições ϵ	22
3.6	Hipervolume gerado pelas soluções da fronteira de P , ilustrado aqui pela região em cinza.	29
3.7	Fluxograma básico do algoritmo SPEA2	30
3.8	Algoritmo de corte	32
4.1	Exemplo ilustrativo de distribuição geográfica de fornecedor e clientes	34
4.2	Estoque inicial e capacidade de estoque de cada cliente	34
4.3	Rota para um dia da solução de reabastecimento de estoque e quantidade abastecida para cada cliente	35
4.4	Níveis de estoque no dia seguinte	36
4.5	Rotas dos dias do horizonte de planejamento. À esquerda, são apresentadas todas as rotas diárias em um gráfico único. À direita são apresentadas as mesmas rotas, mas uma para cada dia do horizonte de planejamento.	36
4.6	Estrutura de dados de uma instância de teste	44
5.1	Codificação do genótipo	50
5.2	Recombinação vertical de um ponto	54
5.3	Recombinação horizontal de um ponto	55
5.4	Recombinação uniforme de células	56
5.5	Recombinação uniforme de colunas	57
5.6	Recombinação uniforme de linhas	58

5.7	União de reabastecimentos	60
5.8	Inserção de reabastecimento	61
5.9	Troca de reabastecimentos	62
5.10	Remoção de reabastecimentos	64
5.11	Deslocamento de reabastecimento	65
5.12	Troca de reabastecimento em rota de alto custo	66
5.13	Alocação máxima em clientes de baixo custo de estoque	67
5.14	União e separação de rotas I	68
5.15	Rotas antes e depois da aplicação do operador	69
5.16	União e separação de rotas II	70
5.17	Rotas antes e depois da aplicação do operador	71
5.18	Módulos do algoritmo empregado para otimização multiobjetivo	72
6.1	Dados e solução da instância 5, com horizonte de planejamento de 6 dias, custo de estoque alto e 25 clientes	75
6.2	Conjunto de Pareto obtido e solução mono-objetivo para a instância 5, com horizonte de planejamento de 6 dias, custo de estoque baixo e 10 clientes	76
6.3	Conjunto de Pareto e solução mono-objetivo para a instância 4, com horizonte de planejamento de 6 dias, alto custo de estoque e 15 clientes	77
6.4	Amostragem de 6 soluções do conjunto de Pareto da Figura 6.3	78
6.5	Solução mono-objetivo e solução da fronteira de Pareto mais próxima	79
6.6	Conjunto de Pareto inclui a solução mono-objetivo, para a instância 3, com horizonte de planejamento de 3 dias, alto custo de estoque e 15 clientes	80
6.7	Piores resultados	87
6.8	Melhores resultados	88
6.9	Conjunto de Pareto para uma instância com 200 clientes e horizonte de 3 dias	89
6.10	Amostragem de 4 soluções do conjunto de Pareto da Figura 6.9	90
6.11	Conjunto de Pareto para instância com 200 clientes e horizonte de 6 dias	91
6.12	Amostragem de 4 soluções do conjunto de Pareto da Figura 6.11	92

Lista de Tabelas

2.1	Critérios de classificação para problemas de estoque e roteamento	11
6.1	Resultados para instâncias com $H = 3$ e $h_i \in [0, 01; 0, 05]$	82
6.2	Resultados para instâncias com $H = 3$ e $h_i \in [0, 1; 0, 5]$	83
6.3	Resultados para instâncias com $H = 6$ e $h_i \in [0, 01; 0, 05]$	84
6.4	Resultados para instâncias com $H = 6$ e $h_i \in [0, 1; 0, 5]$	85

Lista de Algoritmos

1	Função de dominância	49
2	Cálculo de alocação	52

Capítulo 1

Introdução e Motivação

1.1 Introdução

VMI (do inglês *Vendor-Managed Inventory*) é uma iniciativa na área de logística integrada, onde o fornecedor monitora o nível de estoque dos clientes e toma decisões de distribuição de acordo com as necessidades dos clientes. Essa prática se popularizou nos anos 80 devido ao sucesso da parceria de um produtor/fornecedor, a Procter & Gamble, com um cliente/distribuidor, o Walmart. Eles projetaram um sistema que interligava a Procter & Gamble aos centros de distribuição do Walmart. Nos centros de distribuição, quando o nível de estoque dos produtos da Procter & Gamble atingia um limiar mínimo, o sistema automaticamente alertava a Procter & Gamble que, por sua vez, planejava a entrega de mais produtos. A economia gerada com o novo sistema de gerenciamento de estoque pôde ser repassada para o Walmart, o qual a repassava aos consumidores finais.

Com base nas informações de estoque de todos os clientes, é possível tratar com eficiência o compromisso entre custos de estoque e nível de serviço ao cliente. Além disso, como o fornecedor também coordena a distribuição, ao invés de responder a demandas recebidas individualmente, os custos de transporte podem ser reduzidos com o VMI, pois é possível promover um melhor planejamento do roteamento de entrega e, conseqüentemente, uma utilização mais eficiente da capacidade de transporte. A classe de problemas que pode obter vantagens com a adoção de uma estratégia VMI é caracterizada pela presença de dois objetivos:

- otimização dos custos de estoque;
- otimização da utilização da capacidade de transporte.

Os casos em que esses dois aspectos são considerados simultaneamente constituem os problemas de estoque e roteamento (IRP, do inglês *Inventory Routing Problem*).

De forma geral, pode-se dizer que um IRP compreende três decisões:

- quais clientes atender em cada instante de tempo;
- quando um cliente é visitado, qual a quantidade a ser fornecida;
- qual rota deve ser seguida para cada conjunto de clientes a serem atendidos.

Pode-se identificar, no problema de estoque e roteamento gerenciados pelo fornecedor, dois problemas de otimização combinatória: (i) o problema de alocação, que envolve as determinações de quais clientes devem ser atendidos e qual o montante a ser reabastecido em cada cliente; (ii) o problema de roteamento de veículos.

O objetivo desse trabalho é apresentar uma proposta de solução para uma das formulações de um IRP disponível na literatura. Segundo a formulação, um fornecedor é responsável por suprir a demanda de um determinado número de clientes, em um certo horizonte de tempo, utilizando um veículo com capacidade limitada. Os clientes apresentam restrições de estoque máximo e não podem ficar desabastecidos. O fornecedor possui capacidade de produção também limitada. Os custos envolvidos compreendem os custos de estoque no fornecedor e nos clientes e os custos de transporte para a distribuição dos produtos. Na formulação encontrada na literatura, o objetivo é minimizar a soma dos custos de estoque e transporte, o que conduz a um problema de otimização mono-objetivo. Dada a natureza essencialmente multiobjetivo do problema original, é proposta neste trabalho uma formulação multiobjetivo que considera explicitamente os custos de estoque e transporte.

Existem várias abordagens para tratar problemas de otimização combinatória como o IRP. Apesar da melhoria de desempenho significativa dos softwares que implementam métodos de otimização exata, o problema de escalabilidade está sempre presente. Quando se aumenta o número de variáveis e restrições, ocorre uma explosão combinatória que resulta em um elevado tempo computacional para gerar uma solução exata. Dependendo do tamanho do problema, o custo computacional pode ser proibitivo. Esse aspecto faz com que o uso de heurísticas se torne uma opção muito atraente para a resolução de problemas de grande porte. Nos últimos anos, tem havido um interesse crescente por métodos heurísticos capazes de encontrar soluções de alta qualidade, não necessariamente ótimas, através da aplicação de processos iterativos. Nesse tipo de abordagem, o foco principal é no comportamento de convergência do processo iterativo. São três as condições desejáveis: (i) garantia de convergência, (ii) tempo de convergência compatível com as necessidades de cada aplicação, e (iii) convergência para a solução ótima. Quanto mais complexo o problema de otimização e quanto maior a dimensão do espaço de busca, menor a chance de se terem atendidas estas três condições simultaneamente.

1.2 Motivação da pesquisa

Os problemas de estoque e roteamento gerenciados pelo fornecedor apresentam, tipicamente, vários objetivos. Objetivos esses muitas vezes conflitantes entre si: atender o cliente o mais rápido possível, usar de forma eficiente a frota de transporte disponível, minimizar o estoque onde os custos são altos, entre outros. Essa característica torna essa classe de problemas propícia para a aplicação de otimização multiobjetivo.

O propósito principal da otimização multiobjetivo é encontrar soluções não-dominadas da fronteira de Pareto, denominadas soluções Pareto-ótimas, onde cada solução indica, para um dado valor de um dos objetivos, quais são os melhores valores que os demais objetivos podem ter.

São várias as técnicas tradicionais para tratar de problemas multiobjetivos (Deb (2001)):

- Soma ponderada dos objetivos: consiste em criar uma única função-objetivo somando cada objetivo multiplicado por um peso. Os pesos são fornecidos como parâmetros.
- Método de restrições ϵ : um dos objetivos é escolhido para ser minimizado (considerando o problema de minimização), e os demais objetivos $f_m(x)$ são transformados em restrições, tal que $f_m(x) \leq \epsilon_m$. Esse método necessita que a escolha de ϵ_m esteja em uma região factível para cada objetivo.
- Programação por metas: esta técnica tenta encontrar soluções que possam atingir uma meta pré-determinada para uma ou mais funções-objetivo. Caso não exista uma solução factível que alcance as metas para todos os objetivos, tenta minimizar os desvios em relação às metas, sendo que os desvios podem ser positivos ou negativos. Na programação por metas ponderadas, também são atribuídos pesos aos desvios. Ao se resolver os problemas mono-objetivos gerados variando-se as metas e os pesos, obtém-se um conjunto de soluções não-dominadas que se aproximam de soluções Pareto-ótima.

As técnicas apresentadas reduzem o problema de otimização multiobjetivo a um problema de otimização mono-objetivo. Cada técnica usa uma forma diferente de redução e introduz parâmetros adicionais, sendo que a escolha desses parâmetros afeta diretamente os resultados obtidos. Dessa forma, para cada compromisso que se queira entre os objetivos, é necessário escolher os parâmetro adequadamente e solucionar múltiplos problemas de otimização mono-objetivo. Algumas técnicas não garantem soluções ao longo da fronteira de Pareto. Se a fronteira não é convexa, o método do somatório dos pesos não encontra certas soluções, independentemente dos pesos escolhidos. Além disso, a distribuição uniforme desses parâmetros não garante a diversidade das soluções ao longo da fronteira estimada (Deb (2001)).

Em virtude dessas limitações, nos últimos anos vem crescendo o interesse em pesquisas de técnicas alternativas para tratar problemas de otimização multiobjetivo. O uso de meta-heurísticas ganhou rápida aceitação na comunidade científica em uma grande variedade de áreas. Dentre as meta-heurísticas, destacam-se os algoritmos evolutivos (Fogel (1999)), que são baseados na emulação de mecanismos de seleção natural. Algoritmos evolutivos, que eram muito comuns em otimização mono-objetivo, começaram a ser usados em otimização multiobjetivo devido a várias vantagens que eles apresentam. A possibilidade de se trabalhar com várias soluções simultaneamente, de não precisar de informações adicionais, além do valor das funções-objetivo junto a cada solução-candidata, e de poder escapar de ótimos locais fazem dos algoritmos evolutivos técnicas promissoras a serem empregadas nos problemas de otimização multiobjetivo.

A diferença fundamental entre um algoritmo evolutivo tradicional e um algoritmo evolutivo multiobjetivo é o operador de seleção, dado que a comparação entre duas soluções é feita com base no conceito de dominância de Pareto.

A aplicação de um algoritmo evolutivo multiobjetivo apresenta várias vantagens em relação às técnicas tradicionais:

- Não introduz parâmetros adicionais ao problema;
- Trabalha diretamente com várias funções-objetivo usando o conceito de dominância de Pareto;
- Um conjunto diversificado de soluções pode ser encontrado em apenas uma execução do algoritmo;
- É menos suscetível à forma e continuidade da fronteira de Pareto, ou seja, os algoritmos evolutivos tratam facilmente fronteiras de Pareto não-convexas ou que apresentam descontinuidade;
- Não requer que a função-objetivo seja diferenciável.

Essas vantagens tornam o uso de um algoritmo evolutivo multiobjetivo propício para tratar problemas reais, onde um tempo de resposta curto (tempo de execução do algoritmo), diversidade de soluções e necessidade de se considerar objetivos conflitantes são características bastante desejáveis.

São várias as propostas de algoritmos evolutivos voltados para otimização multiobjetivo. Dentre elas, pode-se destacar o MOGA (do inglês, *Multi-Objective Genetic Algorithm*, ver Fonseca & Fleming (1993)), o NSGAII (do inglês, *Non-dominated Sorting Genetic Algorithm*, ver Deb *et al.* (2002)), o SPEA2 (do inglês, *Strength Pareto Evolutionary Algorithm*, ver Zitzler *et al.* (2001)) e o ϵ -MOEA (do inglês, *ϵ -dominance Multi-Objective Evolutionary Algorithm*, ver Deb *et al.* (2003)).

Nesse trabalho, é apresentada uma abordagem baseada no SPEA2, com representação computacional das soluções-candidatas e operadores genéticos concebidos especificamente para tratar da versão

de IRP a ser considerada, combinada com buscas locais. O diferencial da pesquisa, em relação às propostas já existentes na literatura, é a iniciativa de tratar o problema efetivamente como um problema multiobjetivo, usando algoritmos e técnicas concebidas especificamente para esse fim.

1.3 Contribuições da pesquisa

O objetivo do trabalho é propor uma abordagem de otimização multiobjetivo a um problema inerentemente multiobjetivo e encontrado em vários setores da indústria: o problema de estoque e roteamento gerenciados pelo fornecedor.

O uso de um algoritmo comprovadamente eficiente para o tratamento de vários problemas de otimização multiobjetivo não garante a sua eficiência em todos os problemas. O algoritmo pode ser considerado uma "máquina de busca" genérica que necessita de peças específicas para tratar as particularidades de cada problema. São essas peças ou componentes que definem o sucesso ou não do emprego de um algoritmo de otimização.

Neste trabalho, foi feita uma implementação do algoritmo SPEA2 (ver Zitzler *et al.* (2001)) e dos componentes que fazem a integração dele ao problema de otimização de estoque e roteamento gerenciados pelo fornecedor. Em linhas gerais, foram modelados e implementados componentes com as seguintes funções:

- Codificação dos indivíduos da população de forma apropriada para o problema tratado;
- Definição do fenótipo a partir do genótipo;
- Definição do critério de dominância, que é baseado em dominância restrita, permitindo a comparação com/entre soluções factíveis e não-factíveis;
- Operadores genéticos adequados ao tipo de problema: recombinação e mutação;
- Operadores de busca local que usam o conceito de dominância.

Como a etapa de roteamento pode ser executada de forma independente e após a definição da política de reabastecimento, optou-se por empregar uma solução eficiente para o roteamento, já disponível na literatura. Trata-se da heurística de Chained-Lin-Kernighan implementada como parte do projeto Concorde (Concorde Home Page (n.d.)).

Várias instâncias de teste presentes na literatura foram consideradas e os resultados foram comparados com a solução exata fornecida por uma abordagem mono-objetivo existente na literatura. Instâncias de maior porte, geradas artificialmente, também foram consideradas, visando verificar a escalabilidade do algoritmo evolutivo proposto.

1.4 Organização do texto

Esta dissertação encontra-se organizada em 7 capítulos:

- Capítulo 1: Introdução
- Capítulo 2: Problema de estoque e roteamento gerenciados pelo fornecedor
- Capítulo 3: Otimização multiobjetivo
- Capítulo 4: Caracterização do problema
- Capítulo 5: Estratégia de solução
- Capítulo 6: Resultados
- Capítulo 7: Conclusões e perspectivas

O Capítulo 2 oferece uma revisão da literatura sobre o problema de gerenciamento de estoque e roteamento. São descritas diferentes políticas de reabastecimento e as estratégias de solução adotadas.

O Capítulo 3 aborda os conceitos básicos de otimização multiobjetivo, os algoritmos evolutivos em otimização multiobjetivo, as métricas para comparação de desempenho e descreve o algoritmo SPEA2 (do inglês *Strength Pareto Evolutionary Algorithm*), um algoritmo considerado referência em otimização multiobjetivo e que é utilizado na abordagem proposta.

O Capítulo 4 apresenta a definição do problema, as instâncias de teste utilizadas e a formulação multiobjetivo do problema que esse trabalho se propõe a resolver.

O Capítulo 5 descreve a estratégia proposta, detalhes de implementação, a codificação adotada para representar as soluções-candidatas, e os operadores concebidos para mutação, recombinação e busca local.

O Capítulo 6 apresenta e analisa os resultados obtidos, a partir da aplicação da meta-heurística proposta e de comparações com soluções da literatura.

O Capítulo 7 inclui as considerações finais e as perspectivas futuras da pesquisa.

1.5 Síntese do capítulo

Neste capítulo, foram introduzidos os principais conceitos tratados nesse trabalho, voltados para o problema de estoque e roteamento gerenciados pelo fornecedor empregando otimização evolutiva multiobjetivo. Foram apresentados também o escopo, a motivação e os objetivos da pesquisa e a organização do texto. Associadas aos objetivos da pesquisa, também foram mencionadas as principais contribuições alcançadas.

Capítulo 2

Problema de estoque e roteamento gerenciados pelo fornecedor

VMI (do inglês *Vendor-Managed Inventory*) ou Estoque Gerenciado pelo Fornecedor é um processo no qual o fornecedor é responsável pela gestão de estoque dos clientes. Ao invés dos clientes encaminharem seus pedidos de reabastecimento, o fornecedor é quem decide quando e quanto entregar de mercadorias para cada cliente (Campbell *et al.* (2001)).

Esse sistema de gerenciamento de estoques e entregas pelo fornecedor é uma política natural quando os pontos de demanda pertencem à mesma empresa ou grupo que distribui os produtos. Em outros casos, é fruto de negociação entre fornecedores (ou distribuidores) e consumidores, como forma de responder de maneira mais eficiente às variações de demanda e diminuir o risco de falta do produto e outras flutuações indesejáveis ao longo da cadeia de suprimentos. No caso de demanda estocástica, o gerenciamento de estoques pelo fornecedor pode trazer benefícios adicionais, devido ao compartilhamento de um estoque de segurança agregado, o qual é menor que a soma dos estoques de segurança requeridos por cada consumidor, fenômeno conhecido como compartilhamento de risco (do inglês *risk pooling*) (Bramel & Simchi-Levi (1997)). O primeiro passo para o sucesso da técnica VMI requer disponibilidade e acurácia dos dados dos clientes. Para tomar decisões, o fornecedor tem acesso a informações importantes, como níveis de estoques (disponibilidade de produtos) atual e passado de todos os clientes, distância e tempo de viagem dos clientes em relação ao fornecedor e entre os clientes, custo de transporte, custo de manutenção de estoques, custo de falta de estoque (indica baixa qualidade no atendimento ao cliente) e capacidade e disponibilidade de veículos e recursos humanos para a entrega dos produtos (Campbell *et al.* (2001)).

Exemplos de áreas que utilizam o VMI:

- Indústria petroquímica;

- Indústria de varejo (supermercados);
- Indústria de bebidas;
- Indústria automobilística (particularmente distribuição de autopeças);
- Controle de numerário em redes ATM;
- Empresa de coleta de resíduos em estações de tratamento de água;
- Distribuidora de medicamentos e perfumaria;
- Distribuidora de jornais e revistas.

Em muitas aplicações, o fornecedor, além de controlar os estoques dos clientes, também administra uma frota de veículos para transportar os produtos aos clientes. Neste caso, o objetivo do fornecedor é não só administrar o reabastecimento ótimo dos estoques, como também realizar a distribuição dos produtos. Este problema é chamado problema de estoque e roteamento (do inglês *Inventory Routing Problem - IRP*) (Bard *et al.* (1998)).

O objetivo do problema consiste em desenvolver uma estratégia de distribuição que minimize os custos totais de distribuição e estoque, o que representa uma tarefa bem desafiadora. Quanto maior o número de clientes envolvidos, mais difícil torna-se o problema. Se, por um lado, a flexibilidade de decidir quando e quanto entregar reduz significativamente os custos de distribuição e estoque, por outro, esta flexibilidade aumenta a complexidade do problema e dificulta a obtenção de uma boa solução e, mais ainda, da solução ótima de distribuição e estoque.

Na literatura, são tratados vários problemas de estoque e roteamento gerenciados pelo fornecedor com as mais diversas características. As características comuns podem ser consolidadas em uma formulação genérica do problema, apresentada a seguir.

O problema de estoque e roteamento (IRP) trata da distribuição de um ou mais produtos, a partir de um ou mais centros de distribuição, que atendem M clientes dentro de um horizonte de planejamento H . Considera-se que as ações de abastecimento dos clientes se dão em instantes discretos de tempo, dentro do horizonte de planejamento H . Cada cliente s consome o produto a uma taxa r_s e tem uma capacidade de armazenagem U_s . O nível de estoque do cliente s no instante t é $I_{s,t}$. A distribuição dos produtos pode ser feita através de uma frota de veículos homogênea ou heterogênea, e cada veículo tem uma capacidade C_v . A quantidade fornecida ao cliente s no instante t é $x_{s,t}$. Define-se h_s como o custo unitário de estoque do cliente s . O objetivo é minimizar o custo médio diário de distribuição durante o horizonte de planejamento, de modo que não haja indisponibilidade de estoque nos clientes. Pode-se adicionar ao modelo custos de estoque, custos de perdas de vendas (admitindo que pode ocorrer falta de produtos) e até mesmo estimativas de lucro em função dos produtos entregues ou

das vendas. A cada instante t , são tomadas decisões de roteamento de veículos e reabastecimento de estoque dos clientes. Em muitas situações práticas, essas decisões são tomadas diariamente e devem ser executadas ao longo do mesmo dia. Neste caso, cada instante de tempo t está vinculado a um dia.

O custo de uma decisão no instante t pode incluir:

- $c_{i,j}$: custo de transporte dos arcos (i,j) ;
- $L_s(x_{s,t})$: lucro caso o vendedor entregue uma quantidade $x_{s,t}$ ao cliente s no instante t ;
- h_s : custo unitário de estoque no cliente s ;
- $I_{s,t-1}$: nível de estoque do cliente s no dia anterior;
- $x_{s,t}$: quantidade entregue ao cliente s no dia t ;
- r_s : demanda diária do cliente s ;
- $P_s(r_s)$: penalidade de falta se a demanda r_s do cliente s não for atendida. A demanda não atendida é tratada como demanda perdida e não como atraso na entrega.
- $E_{s,t}$: custo de estoque no cliente s no instante t , que pode ser definido como:

$$E_{s,t} = (I_{s,t-1} + x_{s,t} - r_s) * h_s;$$

O problema de estoque e roteamento engloba três decisões:

- Quando atender cada cliente;
- Quanto entregar de mercadoria para cada cliente nas ocasiões de atendimento;
- Qual a melhor rota de entrega, dados os clientes a serem atendidos.

As duas primeiras decisões envolvem a gestão dos estoques nos clientes, devendo ser respeitadas as restrições de capacidade de armazenagem e não ocorrência de desabastecimento de estoque nos clientes. Estabelecidos os dias de atendimento do conjunto de clientes e as quantidades a serem entregues, a definição das rotas de entrega corresponde à resolução de um problema de roteamento de veículos para cada dia do horizonte de planejamento. Uma alternativa de abordagem é resolver esses subproblemas de forma independente e sequencial. Outra abordagem é resolvê-los de forma conjunta, uma vez que a definição dos dias dos atendimentos depende dos custos de distribuição, os quais, por sua vez, dependem da programação dos reabastecimentos.

Em Baita *et al.* (1998), é apresentada uma revisão de uma classe de problemas denominada *DRAI* (do inglês *Dynamic Routing-And-Inventory*). Essa classe de problemas é caracterizada pela presença de três aspectos básicos:

- Roteamento: transporte de mercadorias entre diferentes áreas geográficas, como depósitos e pontos de venda;
- Estoque: quantidade de mercadorias;
- Comportamento dinâmico: a combinação dos dois aspectos, roteamento e estoque, envolvem um terceiro aspecto, o dinâmico, no sentido de que decisões precisam ser tomadas em diferentes momentos dentro do mesmo horizonte de tempo, e uma decisão anterior influencia decisões posteriores.

O problema DRAI trata de como gerenciar a atividade de prover (uma ou mais) mercadorias de (uma ou mais) origens para (um ou mais) destinos durante um horizonte de tempo finito ou infinito, considerando as questões de roteamento e estoque. É proposta uma classificação baseada em sete grupos principais de critérios classificatórios, conforme apresentado na Tabela 2.1.

Tab. 2.1: Critérios de classificação para problemas de estoque e roteamento

<i>Elemento</i>	<i>Atributo</i>	<i>Alternativas</i>		
Topologia da rede de abastecimento	Pontos de abastecimento ou distribuição	Um para um	Um para muitos	Muitos para Muitos
Itens a serem entregues	Número	Um	Muitos	
Demanda	Tipo	Determinística	Estocástica	
	Comportamento	Constante	Variável	
	Distribuição entre clientes	Uniforme	Não-uniforme	
Decisões	Domínio	Frequência de atendimento	Instante de atendimento	
Restrições	Capacidade dos veículos	Mesma capacidade	Capacidade diferente	Não apresenta restrição
	Capacidade de estocagem	Sim	Não	
	Capacidade de abastecimento	Sim	Não	
	Número de veículos	Fornecido	Variável de decisão	Não-restritivo
Custos	Estoque	Manutenção	Falta	Pedido
	Distribuição	Fixo	Proporcional à distância	Proporcional ao número de clientes
Estratégia de solução	Decomposição	Tempo	Agrupa-roteia	
	Agregação	Tempo	Frequência	Distância
	Algoritmo	Exato	Aproximado	
	Programação Matemática	Linear	Inteira	Não-linear

São identificados dois modelos:

- modelo de frequência de atendimento: as variáveis de decisão são as frequências com que os clientes devem ser atendidos. Esse modelo é aplicável a situações estáveis de distribuição, em que tanto a demanda quanto a oferta não são afetadas por sazonalidades ou outros efeitos aleatórios.
- modelo de instante de atendimento: as variáveis de decisão são os instantes de entrega, o horizonte de planejamento é finito e as entregas podem ser realizadas sem obedecer a uma periodicidade fixa.

Uma linha de pesquisa recente, dentre os modelos de instante de atendimento, é o modelo de fluxo dinâmico com demanda desconhecida, em que o problema é tratado como um sistema dinâmico, sujeito a perturbações que, embora desconhecidas, respeitam limites conhecidos *a priori*. Nessa abordagem, o problema passa a ser a determinação de uma estratégia de controle com realimentação que mantenha, ao longo do horizonte de planejamento, o estado do sistema dentro de certos limites pré-estabelecidos.

Em Ball (1998), é apresentada uma análise dos principais modelos e estratégias de resolução de problemas que envolvam simultaneamente decisões de alocação e roteamento. São identificados três tipos de formulações:

- modelos dinâmicos com tamanho de entrega fixo: tratam do caso em que as entregas a um determinado cliente não obedecem a uma periodicidade e, uma vez determinados os dias de atendimento, a quantidade entregue é fixa e calculada em função do cliente e do dia de atendimento;
- modelos dinâmicos com tamanho de entrega variável: a quantidade entregue é uma variável de decisão que deve ser definida juntamente com o dia de atendimento;
- modelos periódicos: as entregas são realizadas obedecendo-se uma periodicidade fixa.

A diferença entre as formulações reside na forma com que são determinadas as quantidades e os dias de atendimento, ou seja, no componente de alocação do problema. As restrições quanto aos dias de entrega ou frequência de atendimento são quase sempre decorrentes de restrições relativas à gestão de estoques.

A coleta de lixo, que envolve clientes dispersos geograficamente e o transporte do lixo a instalações de processamento ou aterros sanitários, é geralmente tratada como um problema de roteamento periódico. A frequência é baseada nas restrições de gestão de estoque, que no caso correspondem à quantidade de lixo que os clientes podem armazenar entre atendimentos sucessivos.

A distribuição de combustíveis e gases industriais trata do reabastecimento repetido de um produto a clientes dispersos geograficamente, de forma a evitar ou minimizar a ocorrência de falta de estoque. É uma gestão de estoque do tipo *push*, em que o distribuidor reabastece seus clientes sem que estes tenham que fazer pedidos. A necessidade de pedidos por parte de clientes ocorre no caso de sistemas de estoque do tipo *pull*.

Venda/distribuição do tipo *driver sell* ocorre tipicamente na distribuição de bebidas e consiste em um ou mais motoristas que visitam clientes dispersos geograficamente, procurando realizar o maior número de vendas possível, sem saber *a priori* a quantidade que cada cliente irá comprar.

Em modelos periódicos e de tamanho de entrega fixo, geralmente são utilizadas heurísticas de duas etapas do tipo agregação-roteamento, seguidas muitas vezes de uma etapa de melhoria baseada em busca local. Em Ball (1998), é apontado também o uso de um problema modificado de caixeiro viajante coletor de prêmios, para a determinação dos dias de atendimento, e do método de relaxação lagrangeana para o caso de modelos dinâmicos com tamanho de entrega variável.

Em Bard *et al.* (1998), é tratado o problema de estoque e roteamento com depósitos satélites. Os veículos de distribuição devem partir de um depósito principal e retornar para o mesmo, mas com a opção de serem reabastecidos ao longo da rota. São dois os objetivos:

- minimizar a distância total percorrida, obedecendo-se a capacidade dos veículos e a duração das viagens;
- maximizar a qualidade de serviço e a fidelidade do cliente.

Apesar do objetivo principal ser a redução dos custos anuais, o planejamento é feito para períodos curtos, mas provendo uma medida de custo de longo prazo que possibilite tomadas de decisão por parte do planejador. Para fins de planejamento, são considerados dois períodos (duas semanas) por vez. São identificados todos os clientes cujo dia de entrega ótima ocorra nesse intervalo de tempo. A escolha de que dia o cliente vai ser atendido é baseada numa combinação do balanço desejado de demanda e da minimização de custos incrementais (custos de atender um cliente em outro dia que não o dia ótimo). Apenas o planejamento do primeiro período é utilizado e um novo planejamento é feito para os dois próximos períodos. Isso resulta em um problema que pode ser resolvido usando programação linear inteira mista com dois critérios. No entanto, essa abordagem só é viável computacionalmente para instâncias pequenas. A proposta apresentada resolve separadamente o problema de alocação e o de roteamento.

Em Chan *et al.* (1998), é proposta a política *Zero Inventory Ordering (ZIO)*, em que cada cliente é abastecido se e somente se o estoque está em zero. Na ausência de restrições de capacidade de veículos ou de frequência em que cada cliente pode ser visitado, a política ZIO é ótima. Existe uma subclasse da ZIO chamada Política de Partição Zero: os clientes são divididos em regiões de tal forma

que cada região é atendida separadamente e independentemente das demais. Quando um cliente é visitado, todos os demais clientes da região são visitados. A proposta é modelar o problema como um *CCLP* (do inglês *Capacitated Concentrator Location Problem*) para gerar a partição de clientes em regiões. Dadas m possíveis localizações para concentradores de capacidade fixa C , o objetivo é achar a localização dos concentradores em um subconjunto dessas m localizações e conectar n terminais, onde cada terminal i usa w_i unidades da capacidade do concentrador, de tal forma que cada terminal seja conectado a um único concentrador, a capacidade do concentrador não seja excedida e o custo total seja minimizado. Existe um custo v_j para o concentrador ser atribuído à localização j e o custo c_{ij} de conectar o terminal i ao concentrador j .

O trabalho apresentado em Bertazzi *et al.* (2005) trata de um problema de produção e distribuição em que uma fábrica produz vários itens que são distribuídos para um conjunto de clientes por uma frota de veículos com capacidade limitada. O problema envolve custo de produção fixa e variável, custo de transporte e estoque no produtor (ou distribuidor) e nos clientes. As decisões englobam quando e quanto produzir, quanto e para quais clientes entregar. O problema é tratado com diferentes políticas de reabastecimento e resolvido através de decomposição em subproblemas. Os subproblemas são resolvidos usando heurísticas e algoritmos exatos.

Em Archetti *et al.* (2007), é proposta uma formulação do problema de estoque e roteamento gerenciados pelo fornecedor. Segundo a formulação, um fornecedor atende diversos clientes em um horizonte de tempo definido, utilizando um único veículo com capacidade limitada para fazer o transporte dos produtos.

O trabalho apresenta uma comparação entre as soluções ótimas do problema, utilizando diferentes políticas de reabastecimento.

São apresentadas três políticas de reabastecimento:

- VMIR-OU (do inglês *vendor-managed inventory routing with order-up*): toda vez que um cliente é visitado, a quantidade entregue é tal que o nível máximo de estoque é atingido;
- VMIR-ML (do inglês *vendor-managed inventory routing with maximum level*): quando o cliente é visitado, a quantidade entregue não pode ultrapassar o nível máximo de estoque, podendo ser qualquer valor até atingir o nível máximo;
- VMIR (do inglês *vendor-managed inventory routing*): quando o cliente é visitado, a quantidade entregue pode ser qualquer valor positivo.

A função-objetivo a ser minimizada é a soma dos custos de estoque e transporte. Mais precisamente, considerando:

- h_0 : custo unitário de estoque no fornecedor;

- B_t : nível de estoque no fornecedor no instante t ;
- h_s : custo unitário de estoque no cliente s ;
- $I_{s,t}$: nível de estoque do cliente s no instante t ;
- $c_{i,j}$: custo de transporte entre os clientes i e j ;
- $y_{i,j}^t$: variável binária igual a um caso o cliente j siga imediatamente o cliente i na rota a ser percorrida no instante t , e zero, caso contrário;
- M : conjunto de clientes;
- $M' = M \cup 0$; conjunto de clientes mais o fornecedor;
- T : horizonte de planejamento composto por H dias;
- $T' = T \cup H + 1$: horizonte de planejamento mais o dia seguinte. O dia seguinte é levado em consideração no cálculo do custo para que seja possível contabilizar as consequências das escolhas realizadas no último dia do horizonte de planejamento.

A função-objetivo a ser minimizada é dada por:

$$\sum_{t \in T'} h_0 B_t + \sum_{s \in M} \sum_{t \in T'} h_s I_{s,t} + \sum_{i \in M'} \sum_{j \in M', j < i} \sum_{t \in T} c_{i,j} y_{i,j}^t \quad (2.1)$$

As soluções ótimas foram obtidas utilizando um algoritmo de ramificação e poda (do inglês *branch-and-cut*). Segundo os resultados obtidos, apesar da política VMIR-OU ser a mais comumente adotada, existe um ganho de eficiência muito maior com as políticas VMIR-ML ou VMIR. Uma grande contribuição do trabalho de Archetti *et al.* (2007) foi a proposta de instâncias de teste para um problema bem definido e a apresentação de soluções ótimas para cada instância.

Em Archetti *et al.* (2011), é apresentada uma nova abordagem de solução para o problema proposto em Archetti *et al.* (2007). A abordagem é baseada em uma heurística que combina busca tabu e modelos de programação inteira mista. A motivação do trabalho foi encontrar uma alternativa viável, do ponto de vista prático, para o problema anteriormente solucionado através de um método exato, mas que apresentava problemas de escalabilidade. A função-objetivo a ser minimizada é a mesma apresentada na formulação 2.1 acrescida de dois termos de penalização relativos à infactibilidade da solução:

$$\sum_{t \in T'} h_0 B_t + \sum_{s \in M} \sum_{t \in T'} h_s I_{s,t} + \sum_{i \in M'} \sum_{j \in M', j < i} \sum_{t \in T} c_{i,j} y_{i,j}^t + \alpha \sum_{t \in T} [Q_t - C]^+ + \beta \sum_{t \in T'} [-B_t]^+ \quad (2.2)$$

onde:

- C : capacidade do veículo de transporte;
- Q_t : quantidade total entregue no dia t ;
- α : fator de penalidade por exceder a capacidade do veículo;
- B_t : nível de estoque no fornecedor no instante t ;
- β : fator de penalidade por faltar produto no fornecedor.
- $[\cdot]^+$: $\max\{\cdot, 0\}$

Em ambos os trabalhos, Archetti *et al.* (2007) e Archetti *et al.* (2011), é usada uma função-objetivo que pondera os diferentes objetivos em um único valor a ser minimizado. A ausência na literatura de propostas de otimização multiobjetivo para o problema proposto nesses trabalhos, associado ao fato de se tratar de um problema inerentemente multiobjetivo e de grande interesse prático, foram as principais motivações para o desenvolvimento do presente trabalho.

2.1 Síntese do capítulo

Neste capítulo, foram apresentados os principais trabalhos na área de IRP, as variações do problema e as soluções propostas. Dentre os trabalhos apresentados, destacam-se Archetti *et al.* (2007) e Archetti *et al.* (2011), que apresentam resultados para o mesmo conjunto de instâncias de teste empregado nos experimentos realizados e descritos no Capítulo 6 desta dissertação.

Capítulo 3

Otimização multiobjetivo

3.1 Introdução

Um problema de otimização multiobjetivo é um problema com dois ou mais objetivos que precisam ser otimizados simultaneamente. É importante mencionar que os objetivos podem ser conflitantes entre si (Coello Coello (1999), Coello Coello (2006)) e que o problema pode estar sujeito a restrições. Isso faz com que o conceito de otimalidade utilizado em otimização mono-objetivo não possa ser utilizado. Em otimização multiobjetivo, o conceito de otimalidade baseia-se na noção introduzida por Francis Ysidro Edgeworth em 1881 (Edgeworth (1881)) e depois generalizada por Vilfredo Pareto em 1896 (Pareto (1896)). É chamado de Edgeworth-Pareto ótimo ou, simplesmente, Pareto-ótimo.

O conceito de solução também difere da otimização mono-objetivo. A solução de um problema multiobjetivo de minimização é formada por um conjunto de soluções que apresentam um compromisso entre os objetivos. Um conjunto de soluções é denominado conjunto Pareto-ótimo se, para cada solução do conjunto, não existe nenhuma outra solução factível capaz de reduzir o valor de um dos critérios do problema sem que simultaneamente cause um aumento em pelo menos um dos demais critérios.

Em otimização multiobjetivo, trabalha-se com dois espaços: o espaço de variáveis e o espaço de objetivos. O espaço de variáveis é onde se faz a busca pelas soluções do problema, ou seja, é o domínio das variáveis do problema. Já o espaço de objetivos é o espaço formado pelas funções-objetivo do problema. A Figura 3.1 mostra um exemplo de mapeamento feito pelas funções-objetivo entre o espaço de variáveis e o espaço de objetivos. No exemplo, o espaço de variáveis é tridimensional e o de objetivos é bidimensional.

Pontos no cubo mostrado no espaço de variáveis não mapeiam necessariamente na região oval do espaço de objetivos. Esses objetos geométricos são apresentados apenas para destacar a dimensionalidade de cada espaço.

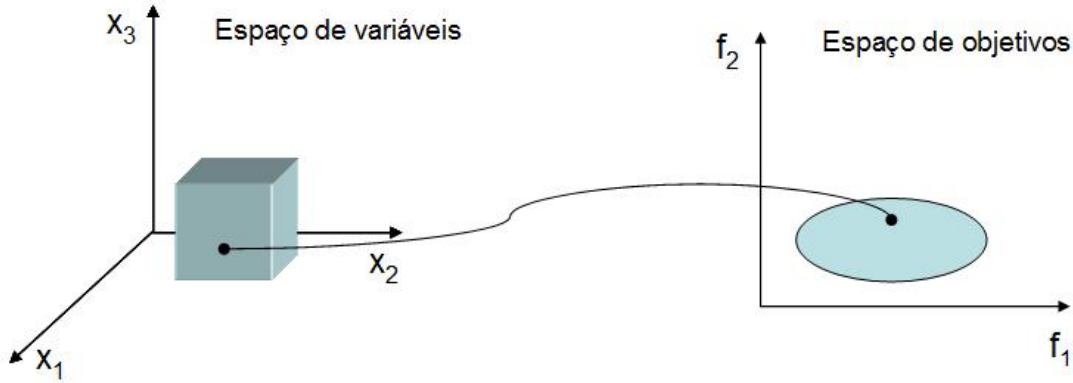


Fig. 3.1: Exemplo de mapeamento de uma solução no espaço de variáveis, tridimensional, para o espaço de objetivos, bidimensional

3.2 Definição

Um problema de otimização multiobjetivo é composto por um conjunto de funções-objetivo a serem otimizadas (maximizadas ou minimizadas) e um conjunto de restrições que devem ser satisfeitas para que a solução seja factível. Supondo a existência de N_{Obj} funções-objetivo que formam o vetor $f(x) = [f_1(x), f_2(x), \dots, f_{N_{Obj}}(x)]^T$, o problema pode ser formulado como:

$$\begin{aligned}
 & \text{maximizar/minimizar} && f(x) \\
 & \text{restrita a} && g_j(x) \geq 0, \quad j = 1, \dots, J; \\
 & && h_k(x) = 0, \quad k = 1, \dots, K; \\
 & && x_i^{(inf)} \leq x_i \leq x_i^{(sup)}.
 \end{aligned} \tag{3.1}$$

onde x é um vetor de variáveis de decisão tal que $x = [x_1, x_2, \dots, x_{N_{var}}]^T$, representando a solução do problema e J e K são, respectivamente, o número de restrições de desigualdade e de igualdade. As desigualdades (g_j) e as igualdades (h_k) são chamadas de funções de restrição e os valores $x_i^{(inf)}$ e $x_i^{(sup)}$ representam os limites inferior e superior para a variável x_i . Esses limites definem o espaço das variáveis. O conjunto de todas as soluções factíveis forma a região factível ou espaço de busca. O vetor de funções-objetivo $f(x) = [f_1(x), f_2(x), \dots, f_{N_{obj}}(x)]^T$ pertencem ao espaço dos objetivos. Para cada solução x no espaço de decisão, existe um ponto $f(x)$ no espaço dos objetivos.

Em otimização multiobjetivo, emprega-se o conceito de dominância de Pareto para comparar duas soluções factíveis do problema. Dadas duas soluções x e y , diz-se que x domina y (representado como $x \preceq y$) se as seguintes condições são satisfeitas (considerando-se um problema de minimização):

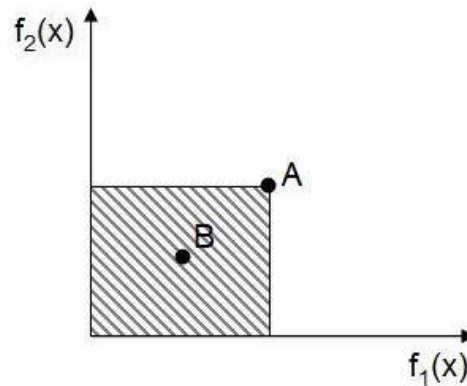


Fig. 3.2: O ponto B, assim como qualquer ponto na região hachurada (desde que corresponda a uma solução factível) domina a solução representada pelo ponto A

- A solução x é melhor ou igual a y em todas as funções-objetivo, ou seja $f_m(x) \leq f_m(y)$ para todo $m = 1, 2, \dots, N_{Obj}$, onde N_{Obj} é o número de funções-objetivo;
- A solução x é estritamente melhor que y em pelo menos uma função-objetivo, ou seja, $f_m(x) < f_m(y)$ para pelo menos um valor de m .

A Figura 3.2 ilustra o conceito de dominância entre pontos de um problema de minimização de objetivos. Nesta figura, $B \preceq A$. Repare que A e B são soluções associadas a valores distintos de x , embora estes pontos estejam sendo plotados no espaço dos objetivos e não no espaço das variáveis.

O conjunto de soluções não-dominadas é chamado de conjunto Pareto-ótimo e representa a solução do problema. A fronteira de Pareto é o conjunto de valores das funções-objetivo das soluções do conjunto Pareto-ótimo.

3.3 Metas da otimização multiobjetivo

Existem três metas importantes em otimização multiobjetivo:

- Obter um conjunto de soluções que esteja o mais próximo possível da fronteira de Pareto;
- Obter um conjunto de soluções com a maior diversidade possível, incluindo aqui critérios como uniformidade no espalhamento ao longo da fronteira de Pareto;
- Alcançar as duas metas anteriores com a maior eficiência computacional possível.

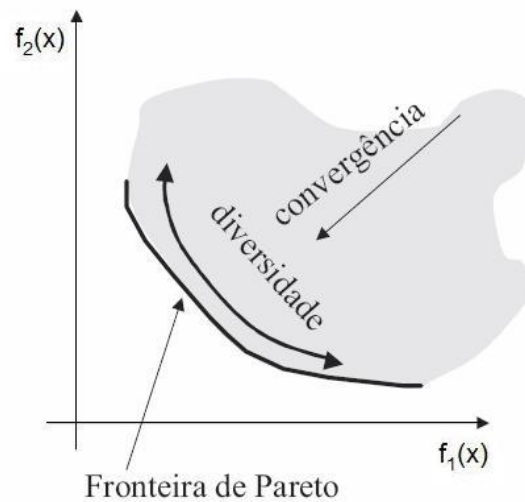


Fig. 3.3: Metas da otimização multiobjetivo

A Figura 3.3 ilustra as duas primeiras metas. É importante notar que a convergência e a diversidade podem ser conflitantes entre si. Portanto, ao avaliar o desempenho de um algoritmo, é necessário usar métricas que considerem ambas as metas.

A primeira meta é comum a qualquer processo de otimização. Por outro lado, encontrar a maior diversidade dentro das soluções é uma meta específica para a otimização multiobjetivo. Como a otimização multiobjetivo trabalha com o espaço de decisão e o espaço dos objetivos, é desejável que as soluções estejam adequadamente distribuídas em ambos os espaços. Dado que encontrar um conjunto de soluções adequadamente distribuídas é uma tarefa que pode consumir consideráveis recursos computacionais, é necessário que tais soluções sejam obtidas eficientemente.

3.4 Abordagens clássicas

Esta seção apresenta algumas das abordagens clássicas usadas para obter a solução de problemas de otimização multiobjetivo.

3.4.1 Soma ponderada dos objetivos

Esse é um dos métodos mais conhecidos e consiste em criar uma função-objetivo somando cada objetivo multiplicado por um peso (Deb *et al.* (2002)). A escolha dos pesos é crucial, pois afeta diretamente a relevância que se atribui a cada objetivo. Requer a normalização de cada função-objetivo, pois elas podem ter diferentes magnitudes. Dessa forma, o problema original, dado pela formulação 3.1, pode ser reformulado, em sua versão para minimização, como:

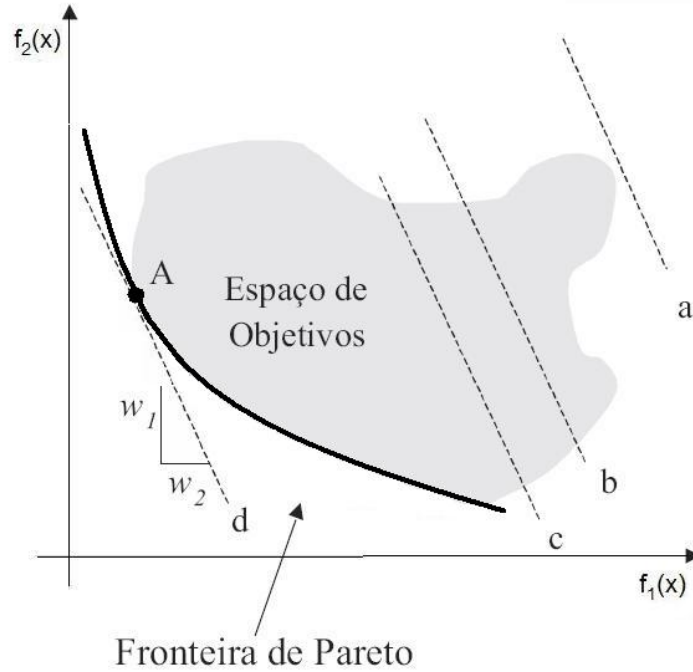


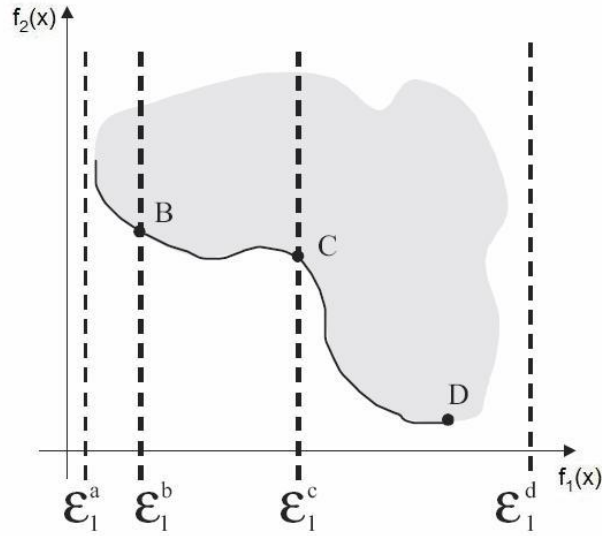
Fig. 3.4: Método de soma ponderada dos objetivos

$$\begin{aligned}
 &\text{minimizar} && f(x) = \sum_{m=1}^{N_{Obj}} w_m f_m(x); \\
 &\text{restrita a} && g_j(x) \geq 0, \quad j = 1, \dots, J; \\
 &&& h_k(x) = 0, \quad k = 1, \dots, K; \\
 &&& x_i^{(inf)} \leq x_i \leq x_i^{(sup)}.
 \end{aligned} \tag{3.2}$$

onde $w_m \in [0, 1]$ é o peso de cada função-objetivo f_m ($m = 1, \dots, N_{Obj}$).

A Figura 3.4 mostra o espaço de objetivos, a fronteira de Pareto e o vetor de pesos $w = (w_1, w_2)$ para um problema com dois objetivos. Para este vetor de pesos, a solução obtida é dada pelo ponto A indicado na figura.

No caso de problema multiobjetivo em regiões não-convexas à esquerda da fronteira de Pareto, este método não é capaz de determinar todas as soluções. Além disso, a aplicação de vetores de pesos uniformemente distribuídos não garante que seja obtido um conjunto de soluções uniformemente distribuídas no espaço dos objetivos (Deb (2001)).

Fig. 3.5: Método de restrições ϵ

3.4.2 Método de restrições ϵ

Esse método foi introduzido em Haimes *et al.* (1971). Um dos objetivos é escolhido como o único objetivo a ser otimizado, com os demais sendo tratados como restrições de desigualdade do problema, permitindo que os métodos clássicos de resolução para otimização mono-objetivo possam ser aplicados. Dessa forma, o problema apresentado na formulação 3.1, em sua versão para minimização, pode ser reformulado como:

$$\begin{aligned}
 &\text{minimizar} && f_u(x), \quad u \in \{1, \dots, N_{Obj}\} \\
 &\text{restrita a} && f_m(x) \leq \epsilon_m, \quad m = 1, \dots, N_{Obj} \text{ e } m \neq u; \\
 &&& g_j(x) \geq 0, \quad j = 1, \dots, J; \\
 &&& h_k(x) = 0, \quad k = 1, \dots, K; \\
 &&& x_i^{(inf)} \leq x_i \leq x_i^{(sup)}.
 \end{aligned} \tag{3.3}$$

onde cada ϵ_m , definido pelo usuário, representa um limite máximo para o valor de f_m . Por exemplo, para um problema de otimização multiobjetivo não-convexo com dois objetivos f_1 e f_2 , escolhe-se f_2 para ser minimizado e mantém-se f_1 com a restrição $f_1 \leq \epsilon_1$.

A Figura 3.5 mostra o espaço dos objetivos e vários valores para ϵ_m . O mínimo de f_2 depende da escolha de ϵ . Por exemplo, usando ϵ_1^c o valor mínimo para f_2 está associado ao ponto C . Empregando-se diferentes valores de ϵ , encontram-se diferentes soluções Pareto-ótimas.

A vantagem dessa abordagem é que pode ser usada em problemas com espaço de objetivos con-

vexo, não-convexo ou discreto. No entanto, necessita que a escolha do vetor ϵ esteja em uma região factível para cada objetivo. Assim como na soma ponderada de objetivos, não há garantia de se conseguir um bom espalhamento do conjunto de soluções, mesmo usando uma distribuição uniforme de valores de ϵ .

3.4.3 Programação por metas

Em problemas práticos, muitas vezes é difícil e até impossível atingir os objetivos e ainda satisfazer uma série de restrições. Nesses casos, é razoável que os gestores procurem satisfazer ou aproximar-se dos objetivos e restrições estabelecidos, ao invés de considerá-los como rígidos.

A programação por metas é uma técnica que permite a modelagem e a busca de soluções para problemas com múltiplos objetivos ou metas a serem otimizadas (Deb (2001)). Não se busca somente maximizar ou minimizar a função-objetivo diretamente, busca-se também a minimização dos desvios no alcance das metas. Para isso, são utilizadas as chamadas variáveis de desvio. O propósito é a minimização dos desvios (distância) em relação aos objetivos estabelecidos, sendo que a solução ideal é aquela na qual todas as variáveis de desvio possuem valor igual a zero, ou seja, em que todos os objetivos são atingidos em seus parâmetros estipulados.

A função-objetivo pode ser formulada como:

$$\min \sum_{i=1}^k |f_i(x) - T_i|, \quad \text{sujeito a } x \in \mathcal{F} \quad (3.4)$$

onde T_i é a meta estabelecida pelo gestor para a i -ésima função-objetivo $F_i(x)$, e \mathcal{F} representa a região de factibilidade. O objetivo é minimizar a soma dos valores absolutos das diferenças entre as metas e os valores obtidos.

Em Romero (2004), são descritas três formas tradicionais de programação por metas: baseada em pesos, também chamada de programação por metas arquimediana; e programação por metas lexicográfica ou programação por metas não-arquimediana e programação por metas minmax, também conhecida como programação por metas nebulosa.

3.4.4 Análise das abordagens tradicionais

Os algoritmos tradicionais para a solução de problemas multiobjetivo apresentam algumas dificuldades, principalmente quando se tem o interesse de encontrar múltiplas soluções Pareto-ótimas:

- Normalmente apenas uma solução Pareto-ótima é encontrada a cada execução de um algoritmo

tradicional;

- Nem todas as soluções da fronteira de Pareto podem ser encontradas por alguns algoritmos, particularmente em problemas multiobjetivo com fronteira de Pareto não-convexa;
- Todos os algoritmos exigem alguma forma de conhecimento sobre o problema, tais como pesos adequados, valores de restrições ϵ e metas para os objetivos.

Diante disso, uma abordagem que tem recebido grande atenção, e vem sendo aplicada com sucesso nos últimos anos, são os algoritmos evolutivos para otimização multiobjetivo (do inglês Multi-Objective Evolutionary Algorithms - MOEAs) (Deb (2001), Coello Coello (2006)), os quais serão apresentados na próxima subseção.

3.5 Algoritmos evolutivos para otimização multiobjetivo

Vários autores destacam o emprego de algoritmos evolutivos em problemas de otimização multi-objetivo pelos seguintes fatores:

- Apresentam facilidade e flexibilidade de modelagem;
- Exigem um menor esforço para encontrar cada solução do conjunto de soluções não-dominadas, se comparado ao esforço de encontrar uma solução para uma formulação mono-objetivo;
- São menos susceptíveis às características da fronteira de Pareto (não convexa, descontínua);
- Podem trabalhar em espaços de busca que são intratáveis pelas abordagens tradicionais.

Os algoritmos evolutivos de otimização multiobjetivo mais relevantes, desenvolvidos entre 1989 e 1998, são:

- NSGA (do inglês, *Nondominated Sorting Genetic Algorithm*): proposto em Srinivas & Deb (1994), é baseado em vários níveis de classificação dos indivíduos. Todos os indivíduos não-dominados são classificados em uma categoria. Esse grupo de indivíduos classificados é removido e outro nível de indivíduos não-dominados é considerado. O processo continua até que todos os indivíduos da população sejam classificados. Esse algoritmo não é muito eficiente porque requer um número elevado de repetições de classificação.
- NPGA (do inglês, *Niched-Pareto Genetic Algorithm*): proposto em Horn *et al.* (1994), usa seleção por torneio baseado na relação de dominância. Dois indivíduos são escolhidos aleatoriamente e comparados com um subconjunto da população (tipicamente 10% da população).

Se um deles é dominado por esse subconjunto e o outro não, então o indivíduo não-dominado ganha. No caso dos dois serem dominados ou dos dois dominarem, o resultado do torneio é decidido através de compartilhamento de *fitness*. O objetivo do compartilhamento de *fitness* é distribuir a população ao longo dos picos do espaço de busca. Cada pico recebe uma parte da população proporcional à altura do pico.

- MOGA (do inglês, *Multi-Objective Genetic Algorithm*): proposto em Fonseca & Fleming (1993), a classificação de um indivíduo corresponde ao número de indivíduos na população corrente que o dominam. A todos os indivíduos não-dominados é atribuído o maior valor possível de *fitness*, enquanto os dominados são penalizados de acordo com a densidade da população na região em que se encontram.

Esses algoritmos, baseados em idéias simples, são considerados a primeira geração de algoritmos evolutivos para otimização multiobjetivo.

A segunda geração de algoritmos evolutivos começou quando o elitismo passou a ser considerado. Esse elitismo geralmente se refere ao uso de uma população externa para manter os indivíduos não-dominados, encontrados ao longo do processo evolutivo. A motivação surgiu do fato de uma solução que não é dominada por soluções-candidatas de uma população corrente não necessariamente ser não-dominada em relação a todas as soluções-candidatas já produzidas pelo algoritmo. Os principais algoritmos são:

- SPEA (do inglês *Strength Pareto Evolutionary Algorithm*) e SPEA2 (do inglês *Strength Pareto Evolutionary Algorithm 2*): o SPEA2 é uma evolução do SPEA (Zitzler & Thiele (1999)) e está descrito mais detalhadamente na seção 3.7, por ter sido o algoritmo escolhido para a abordagem proposta nesse trabalho.
- PAES (do inglês *Pareto Archived Evolution Strategy*): proposto em Horn *et al.* (1994), consiste em uma estratégia evolutiva (1+1) (Beyer (2001)), ou seja, um pai gera um único filho, que emprega busca local e usa um arquivo externo que armazena soluções não-dominadas encontradas até o momento. Esse arquivo é usado como referência para obter uma classificação de dominância aproximada para os candidatos que sofreram mutação.
- NSGA-II (do inglês, *Nondominated Sorting Genetic Algorithm II*): proposto em Deb *et al.* (2002) como uma melhoria do NSGA. Ele estima a densidade das soluções vizinhas a uma solução da população através do cálculo da distância de agrupamento (do inglês, *crowding distance*). A distância de agrupamento de uma determinada solução i é uma estimativa do tamanho do maior cubóide que engloba a solução i sem incluir uma ou mais dentre as demais soluções da população. Essa estimativa de tamanho usa a média do comprimento dos lados.

Ele não usa uma população ou arquivo externo, sendo que o mecanismo de elitismo consiste em combinar os melhores pais com os melhores filhos obtidos. Esse algoritmo é muito mais eficiente que seu predecessor e é considerado um algoritmo de referência.

3.6 Métricas de desempenho

A comparação de desempenho de um ou vários métodos de otimização multiobjetivo é uma tarefa complexa. Conforme citado anteriormente, duas metas da otimização multiobjetivo são: a convergência e a diversidade das soluções encontradas. Portanto, são necessárias pelo menos duas métricas para avaliar ambos os algoritmos, uma métrica para avaliar a convergência e outra para avaliar a diversidade das soluções encontradas.

As métricas de convergência podem comparar soluções P encontradas com o conjunto de soluções Pareto-ótimas do problema, denominado P' . Também é possível usar essas métricas para comparar (de forma relativa) soluções P_1 produzidas por um algoritmo com soluções P_2 produzidas por outro algoritmo.

1. Taxa de Erro

Calcula o número de soluções em P que não estão em P' :

$$ER (Error Ratio) = \frac{|i \in P \cap P'|}{|P|} \quad (3.5)$$

onde $|\cdot|$ indica a cardinalidade de um conjunto de elementos.

Quanto menor for o valor de ER , melhor será a convergência. Se $ER = 0$, significa que $P \subseteq P'$. É aplicável apenas a problemas de otimização discreta.

2. Distância Geracional

Representa a distância euclidiana média (no espaço dos objetivos) entre as soluções de P e P' :

$$GD (Generational Distance) = \frac{\sum_{i \in P} mindist_i}{|P|} \quad (3.6)$$

onde $mindist_i$ representa a distância euclidiana no espaço dos objetivos entre a solução i e a solução mais próxima dela no conjunto P' :

$$mindist_i = \min_{\{j \in P'\}} \sqrt{\sum_{m=1}^{N_{obj}} [f_m(i) - f_m(j)]^2} \quad (3.7)$$

Quanto mais próximo de zero for o valor de GD , melhor será a convergência de P . Pode ser aplicado a problemas de otimização discreta e contínua.

3. Métrica de Cobertura

Dados dois conjuntos de soluções P e Q , essa métrica calcula a proporção de soluções de Q que são fracamente dominadas pelas soluções de P :

$$SC(P, Q) \text{ (Set Coverage)} = \frac{|\{i \in P | \exists j \in Q \text{ e } i \preceq j\}|}{|Q|} \quad (3.8)$$

Quando $SC(P, Q) = 1$ todas as soluções de Q são dominadas por soluções de P . Se $SC(P, Q) = 0$, então nenhuma solução de Q é fracamente dominada pelas soluções em P .

As métricas de diversidade, por sua vez, calculam a distribuição das soluções de um conjunto P .

1. Espaçamento

Calcula o desvio-padrão entre as distâncias de soluções consecutivas (no espaço dos objetivos) do conjunto P :

$$SP \text{ (Spacing)} = \sqrt{\frac{1}{|P| - 1} \sum_{i \in P} (neardist_i - \overline{neardist})^2} \quad (3.9)$$

onde $neardist_i$ representa a distância euclidiana (no espaço dos objetivos) entre a solução i e a solução mais próxima dela no conjunto P , sendo expresso como:

$$neardist_i = \min_{j \in P, j \neq i} \sqrt{\sum_{m=1}^{N_{obj}} [f_m(i) - f_m(j)]^2} \quad (3.10)$$

O valor $\overline{neardist}$ é a média dos valores $neardist_i$:

$$\overline{neardist} = \frac{\sum_{i \in P} neardist_i}{|P|} \quad (3.11)$$

Quanto menor for o valor da métrica SP , melhor distribuídas estão as soluções do conjunto P .

2. Número de nichos

Essa métrica calcula o número de nichos dentro de um conjunto de soluções P :

$$NC \text{ (Niche Count)} = \frac{1}{|P| - 1} \sum_{i, j \in P} |\{j | dist_{ij} > \sigma\}| \quad (3.12)$$

onde $dist_{ij}$ é a distância entre as soluções i e j do conjunto P . O valor NC representa o número de soluções cuja distância entre elas é maior que o parâmetro σ . O valor de σ é atualizado dinamicamente. Quando $dist_{ij} < \sigma$, as soluções i e j estão no mesmo nicho. Quanto maior a quantidade de nichos formados em P , melhor distribuídas estão as soluções.

3. Espalhamento

Avalia a dispersão das soluções no conjunto P ao longo da fronteira de Pareto, assim como a distribuição entre soluções contíguas (no espaço dos objetivos) de P :

$$SPREAD = \frac{\sum_{m=1}^{N_{obj}} extdist_m + \sum_{i \in P} |neardist_i - \overline{neardist}|}{\sum_{m=1}^{N_{obj}} extdist_m + |P| \times \overline{neardist}} \quad (3.13)$$

onde $extdist_m$ representa a distância euclidiana entre as soluções extremas na m -ésima função-objetivo dos conjuntos P e P' :

$$extdist_m = dist_{ij} | i = \min_{k \in P} f_m(k) \text{ e } j = \min_{k \in P'} f_m(k) \quad (3.14)$$

no caso de minimização da função f_m . O valor ideal para $SPREAD$ é 0.

Atualmente, uma métrica bastante usada na avaliação de algoritmos multiobjetivos é o indicador de hipervolume. Calcula-se o volume da região coberta entre os pontos das soluções do conjunto P encontradas e um ponto de referência. Para cada solução i pertencente a P , é construído um hipercubo v_i com referência a um ponto W . O ponto de referência pode ser encontrado construindo-se um vetor com os piores valores de função-objetivo. O resultado da métrica é a união de todos os hipercubos encontrados. Quanto maior o valor do hipervolume melhor, pois um alto valor de hipervolume indica que houve um elevado espalhamento entre as soluções de P e indica também que houve uma melhor convergência. A Figura 3.6 ilustra o cálculo de hipervolume. Essa métrica é calculada através de:

$$HV(Hipervolume) = \sum_{i \in P} v_i \quad (3.15)$$

onde v_i representa o hipercubo da solução i .

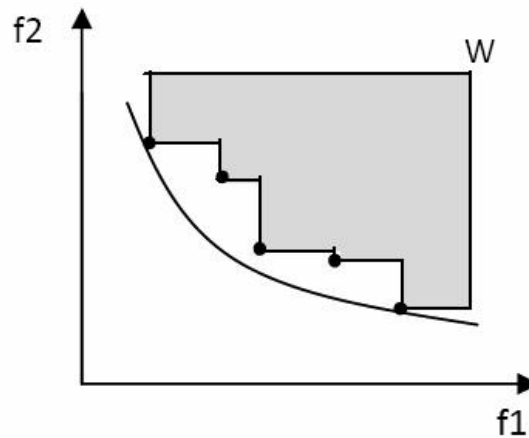


Fig. 3.6: Hipervolume gerado pelas soluções da fronteira de P , ilustrado aqui pela região em cinza.

3.7 SPEA2

O SPEA2 (do inglês *Strength Pareto Evolutionary Algorithm 2*) proposto em Zitzler *et al.* (2001) é um método evolutivo de otimização multiobjetivo que inclui o conceito de elitismo.

A Figura 3.7 mostra o fluxograma básico do algoritmo. O método emprega uma população P_t e um arquivo externo \overline{P}_t . No arquivo externo \overline{P}_t , são armazenadas as soluções não-dominadas até o momento. São fornecidos como parâmetros iniciais do algoritmo o tamanho da população P_t , denominado N , o tamanho do arquivo externo \overline{P}_t , denominado \overline{N} , e o número máximo de gerações, dado por T .

O algoritmo inicia criando uma população inicial P_0 e um arquivo externo \overline{P}_0 inicialmente vazio. A cada iteração, a função de *fitness* (ver seção 3.7.1) é calculada para cada solução $i \in P_t \cup \overline{P}_t$. Em seguida, executa-se a seleção: as soluções não-dominadas de $i \in P_t \cup \overline{P}_t$ são copiadas para o novo arquivo externo \overline{P}_{t+1} . Se o conjunto de soluções não-dominadas tem o tamanho exato do arquivo externo, a seleção está finalizada. Senão, pode-se ter duas situações:

- Tamanho do arquivo externo menor que o número de soluções não-dominadas. Neste caso, ordenam-se as soluções de $P_t \cup \overline{P}_t$, de acordo com a função de *fitness*, e preenche-se o arquivo externo com as soluções dominadas com maior valor da função de *fitness*.
- Tamanho do arquivo externo excede o tamanho \overline{N} . Nesse caso, é usado um algoritmo de corte descrito na subseção 3.7.2.

Por último, realiza-se o processo de seleção por torneio com reposição, recombinação e mutação sobre \overline{P}_{t+1} para se obter a nova população P_{t+1} .

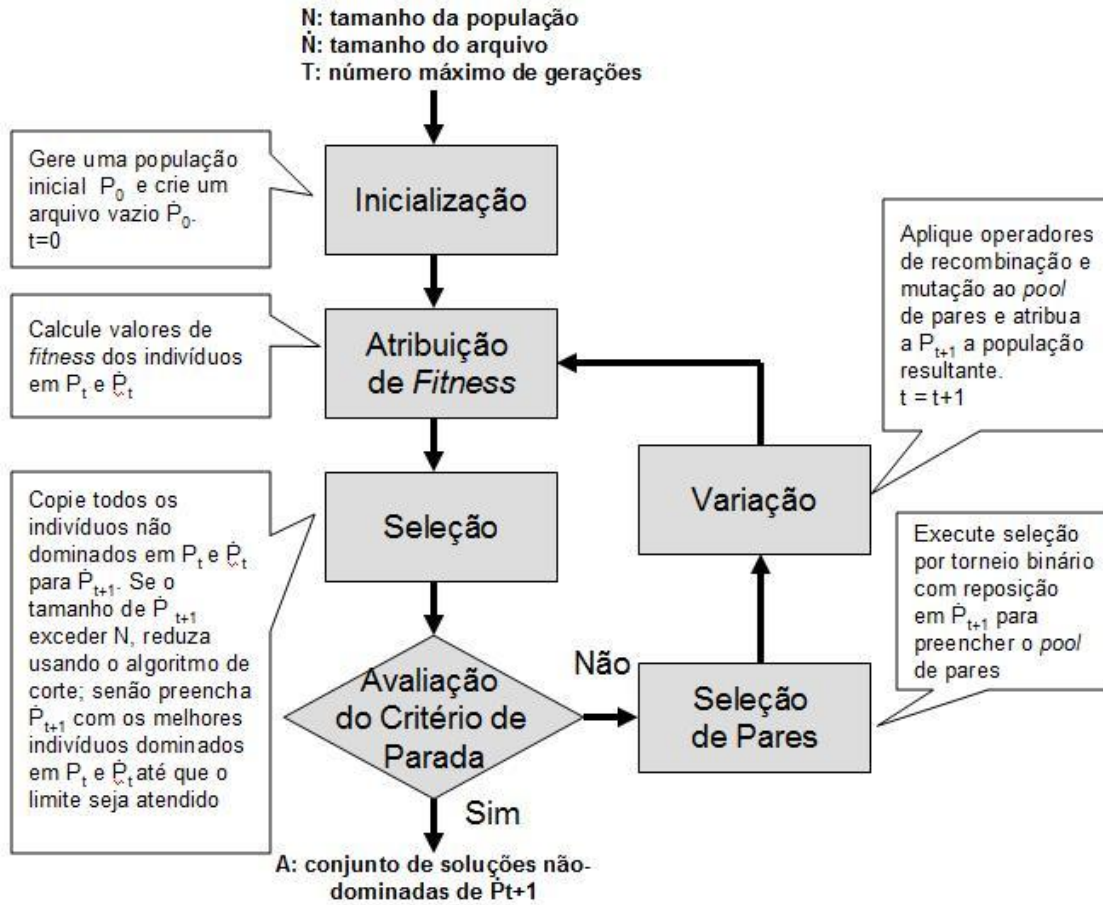


Fig. 3.7: Fluxograma básico do algoritmo SPEA2

3.7.1 Função de *fitness*

No cálculo da função de *fitness*, são usados os conceitos de dominância e de densidade. O objetivo é minimizar o valor da função de *fitness*. Quanto menor o valor da função de *fitness* de um indivíduo, melhor é a adaptação do indivíduo.

A força do indivíduo é dada pelo número de soluções que ele domina:

$$S(i) = |\{j | j \in P_t + \bar{P}_t \wedge i \succ j\}| \quad (3.16)$$

onde $|\cdot|$ é a cardinalidade do conjunto, $+$ define a união de conjuntos e o símbolo \succ corresponde à relação de dominância de Pareto.

O *fitness* bruto do indivíduo é a soma das forças de todos os indivíduos que o dominam:

$$R(i) = \sum_{j \in P_t + \overline{P}_t, j \succ i} S(j) \quad (3.17)$$

A densidade do indivíduo é uma função decrescente em relação ao k -ésimo vizinho mais próximo. A inclusão de um fator de densidade se deve ao fato de que, nos casos em que existem muitas soluções não-dominadas, $S(i)$ se aproxima de zero para todas as soluções. Nesse caso, é necessário haver um mecanismo para privilegiar soluções dentre as não-dominadas.

$$D(i) = \frac{1}{dist_{ij}(k) + 2} \quad (3.18)$$

Para cada indivíduo i , as distâncias (no espaço dos objetivos) entre i e todos os indivíduos j da população P_t e do arquivo externo \overline{P}_t são calculadas e armazenadas em uma lista. Depois de ordenada a lista em ordem crescente, o k -ésimo elemento representa o termo $dist_{ij}(k)$. É sugerido para k o valor $k = \sqrt{N + \overline{N}}$.

A função de *fitness* adotada pelo algoritmo SPEA2 é dada por:

$$F(i) = R(i) + D(i) \quad (3.19)$$

3.7.2 Algoritmo de corte

O objetivo do algoritmo de corte do SPEA2 é restringir o tamanho de \overline{N}_{t+1} a \overline{N} soluções. Em cada iteração, é removida a solução tal que a sua distância para o seu vizinho mais próximo seja a menor dentre as distâncias existentes. No caso de empate, calcula-se a segunda menor distância, e assim sucessivamente. A solução i é escolhida para ser removida se $i \leq_d j$ para todo $j \in \overline{P}_{t+1}$:

$$i \leq_d j \Leftrightarrow \begin{aligned} & \forall 0 < k < |\overline{P}_{t+1}| : \sigma_i^k = \sigma_j^k \vee \\ & \exists 0 < k < |\overline{P}_{t+1}| : [(\forall 0 < l < k : \sigma_i^l = \sigma_j^l) \wedge \sigma_i^k < \sigma_j^k] \end{aligned} \quad (3.20)$$

onde σ_i^k é a distância entre i e o k -ésimo vizinho mais próximo em \overline{P}_{t+1} . A Figura 3.8 ilustra o algoritmo de corte. À esquerda, é exibido o conjunto de soluções não-dominadas. Considerando $\overline{N} = 5$, são eliminadas soluções até se obter um conjunto de tamanho \overline{N} . À direita, é exibida a

sequência de eliminação de soluções, de acordo com o algoritmo de corte, e o conjunto de soluções não-dominadas resultante.

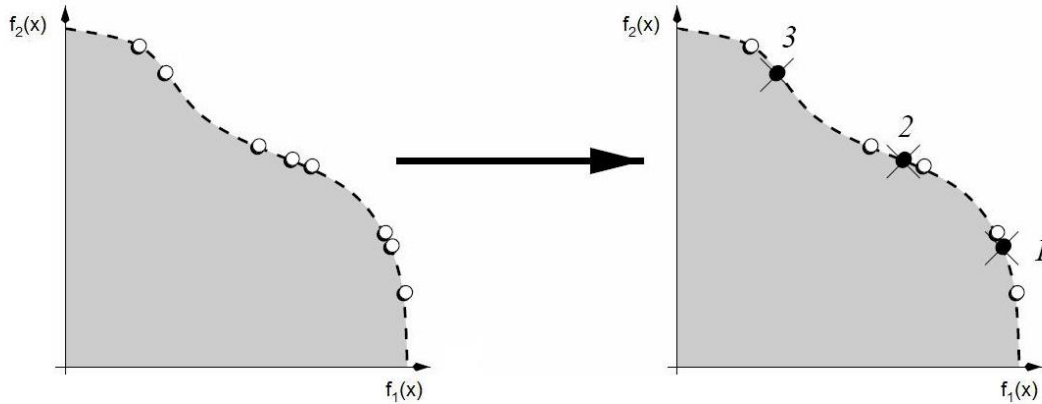


Fig. 3.8: Algoritmo de corte

3.8 Síntese do capítulo

Neste capítulo, foram apresentados conceitos de otimização multiobjetivo, os algoritmos tradicionais na área e os principais algoritmos evolutivos multiobjetivo. Foi descrito também o algoritmo adotado nesse trabalho, o SPEA2. Além disso, foram apresentados os principais critérios usados na comparação entre algoritmos multiobjetivo.

Capítulo 4

Caracterização do problema

4.1 Descrição

O problema de otimização multiobjetivo a ser tratado consiste no planejamento da distribuição de produtos por um fornecedor para um conjunto de clientes em um horizonte de tempo definido. O fornecedor possui um único depósito central e o transporte é feito por um único veículo com uma capacidade de transporte limitada. O veículo pode executar uma viagem por dia durante o horizonte de planejamento.

A Figura 4.1 mostra a distribuição geográfica dos clientes e do depósito central de uma das instâncias de teste a serem consideradas na fase de experimentos. A instância de teste do exemplo compreende 15 clientes e o depósito central pode ser identificado pelo símbolo ■.

Cada cliente possui um estoque inicial. A Figura 4.2 mostra os níveis de estoque inicial dos clientes da instância de teste sendo usada como exemplo. Para cada cliente, a altura total da barra representa a capacidade máxima de estoque, sendo que a parte preenchida da barra representa o estoque inicial e a parte não-preenchida representa a capacidade de estoque ainda disponível.

Uma possível solução do problema consiste em determinar, para cada dia do horizonte de planejamento:

- quais clientes serão atendidos;
- quando um cliente é atendido, qual a quantidade de produto que será entregue;
- qual a rota que será percorrida pelo veículo para realizar as entregas a cada dia do horizonte de planejamento.

A solução para um dos dias do horizonte de planejamento é mostrada na Figura 4.3. O gráfico superior mostra a rota a ser percorrida para atender os clientes selecionados para abastecimento na-

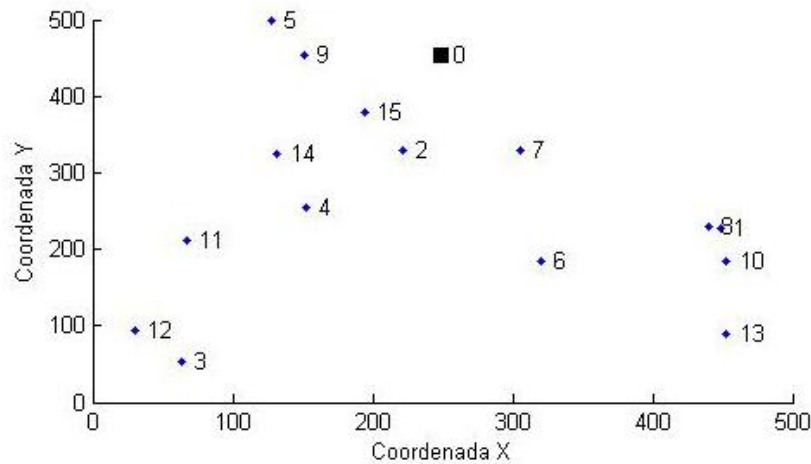


Fig. 4.1: Exemplo ilustrativo de distribuição geográfica de fornecedor e clientes

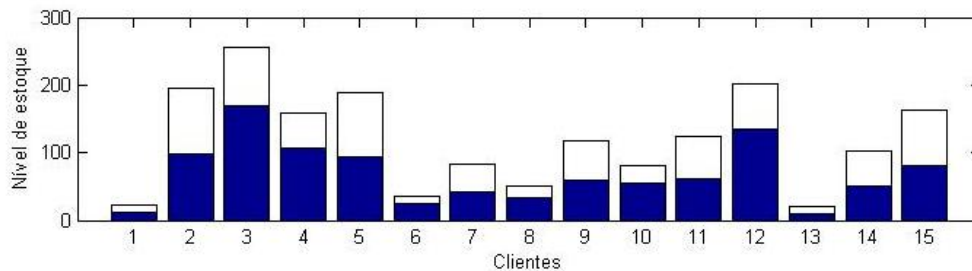


Fig. 4.2: Estoque inicial e capacidade de estoque de cada cliente

quele dia. Não serão abastecidos neste dia os clientes 2, 5, 14 e 15. O veículo deve partir do depósito central, percorrer os clientes e voltar para o depósito. No gráfico inferior, são exibidos os níveis de estoque de cada cliente antes do reabastecimento e os níveis após o reabastecimento. Como o veículo possui capacidade limitada, a solução deve levar em consideração que a soma das quantidades entregues aos clientes em um determinado dia não pode exceder a capacidade máxima do veículo.

Cada cliente apresenta um nível máximo e mínimo de estoque. Na formulação do problema considerado, não existe estoque de segurança, ou seja, o estoque mínimo é sempre igual a zero. No entanto, poder-se-ia facilmente incluir um estoque de segurança, caso necessário, estabelecendo-se valores positivos para o estoque mínimo.

A quantidade a ser abastecida é limitada pelo nível máximo de estoque. Após o abastecimento, o

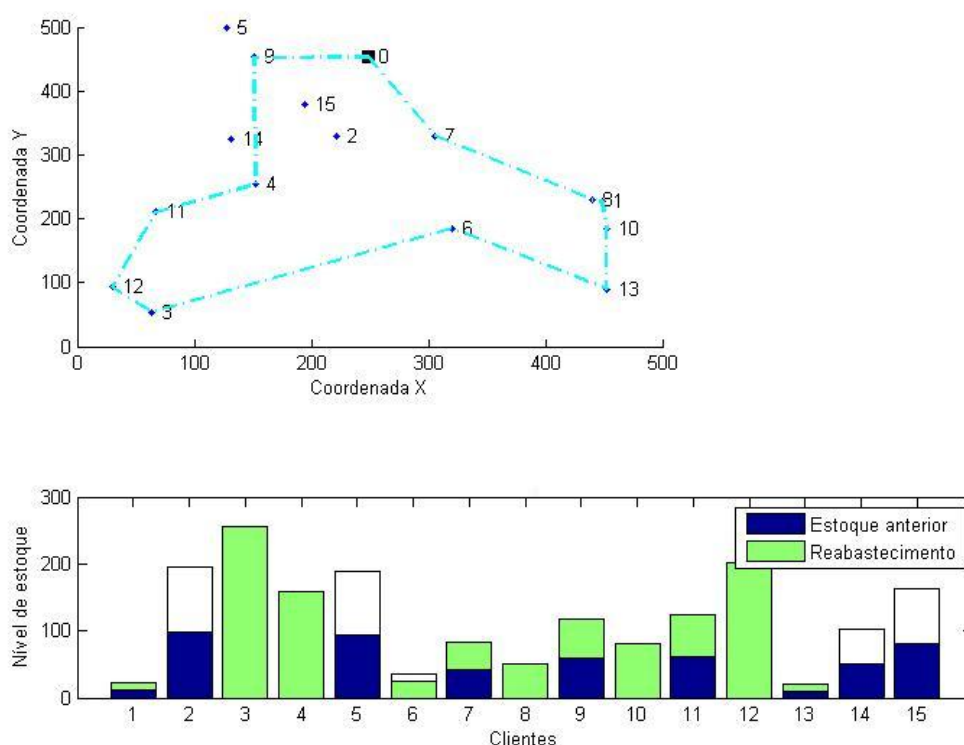


Fig. 4.3: Rota para um dia da solução de reabastecimento de estoque e quantidade abastecida para cada cliente

cliente pode ficar com o estoque abaixo de sua capacidade, como pode-se observar no caso do cliente 6 na Figura 4.3. Se, por um lado, um valor muito baixo de abastecimento pode causar um maior número de abastecimentos e, consequentemente, aumentar o custo de transporte, por outro lado, um valor muito alto de abastecimento pode impactar no custo de estoque.

Cada cliente consome diariamente uma determinada quantidade do estoque de produtos. Na Figura 4.4, é possível observar os níveis de estoque no dia seguinte ao dia exibido na Figura 4.3, já considerando a quantidade consumida no dia. O consumo de cada cliente é constante ao longo do tempo, mas cada cliente pode apresentar um consumo diferente dos demais. Coincidentemente, os clientes 2, 5, 14 e 15, que não foram abastecidos no dia considerado, tiveram seus estoques zerados e deverão ser abastecidos no dia seguinte. Para não violar a restrição de estoque mínimo, só podem ficar sem receber abastecimento clientes que têm estoque suficiente para atender pelo menos a demanda do dia seguinte.

Considerando um horizonte de planejamento de seis dias, a Figura 4.5 exhibe as rotas a serem percorridas em cada dia do horizonte de planejamento. Note que é possível existir dias em que nenhum cliente é atendido. No exemplo dado, no primeiro e sexto dia do horizonte de planejamento

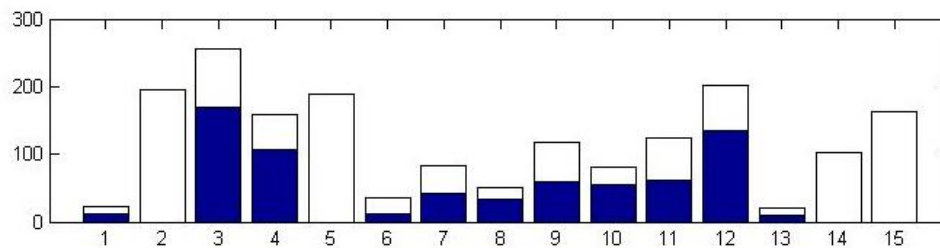


Fig. 4.4: Níveis de estoque no dia seguinte

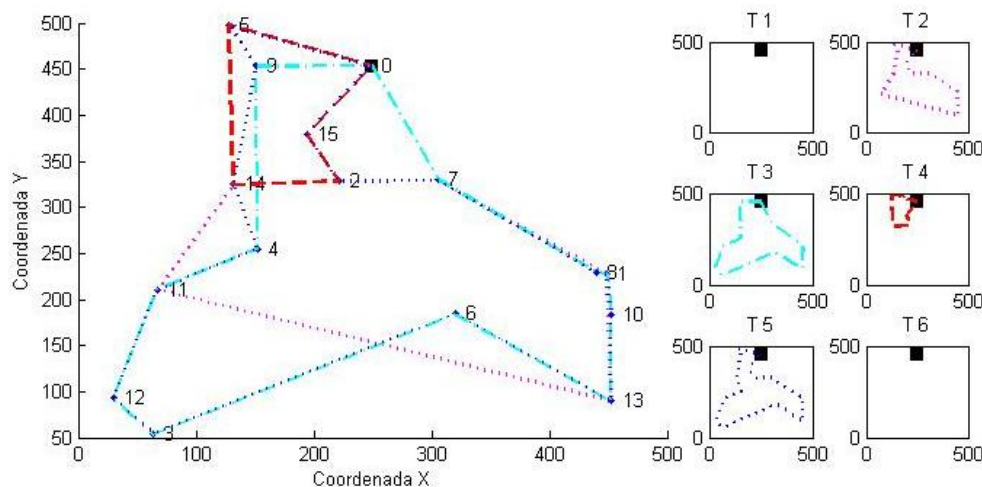


Fig. 4.5: Rotas dos dias do horizonte de planejamento. À esquerda, são apresentadas todas as rotas diárias em um gráfico único. À direita são apresentadas as mesmas rotas, mas uma para cada dia do horizonte de planejamento.

a solução indica que nenhum cliente é atendido. Isso representa uma redução no custo de transporte e vai ocorrer com certa frequência sempre que a capacidade de abastecimento diário superar com alguma folga a demanda diária.

O fornecedor possui um estoque inicial e uma capacidade de produção limitada. A quantidade diária produzida é constante ao longo do horizonte de planejamento. O planejamento de distribuição deve ser feito de modo a não exceder a capacidade de produção do fornecedor.

Existe um custo de transporte que é proporcional à distância percorrida pelo veículo. Existe também um custo unitário de estoque no fornecedor e nos clientes, sendo que os clientes podem apresentar custo de estoque maior ou menor que o do fornecedor. Os custos totais são dados pelo produto entre o custo unitário e a quantidade estocada.

Em relação ao número de objetivos, o problema pode ser resolvido usando

- formulação mono-objetivo;
- formulação multiobjetivo.

Em Archetti *et al.* (2007), é usada uma formulação mono-objetivo, onde o objetivo é minimizar o custo total, composto pelo custo de transporte e de estoque, de acordo com uma ponderação da importância dos dois custos feita *a priori*.

Em casos práticos, é muito comum a existência de mais de um objetivo a ser alcançado. Por exemplo, nos casos em que o fornecedor e os clientes pertencem a empresas diferentes, pode-se desejar separar o que implica em custo para o fornecedor (custo de estoque no fornecedor e custo de transporte) e em custo para o cliente (custo de estoque no cliente). Nesse caso, os objetivos seriam: (i) minimizar custo para o fornecedor e (ii) minimizar custo para o cliente.

Outra situação muito comum é o caso em que o transporte é terceirizado para uma empresa contratada. O uso de uma formulação multiobjetivo em que os dois custos são considerados explicitamente permite que o gestor tenha flexibilidade na negociação dos custos. Esse é o cenário que foi adotado nesse trabalho. O objetivo, nesse caso, passa a ser: (i) minimizar a soma dos custos de estoque (no fornecedor e nos clientes) e (ii) minimizar o custo de transporte.

4.2 Classificação

Devido à grande variedade de problemas que envolvem estoque e roteamento na literatura, a classificação do problema torna-se importante para situá-lo e permitir a comparação entre os problemas.

Seguindo a Tabela 2.1, proposta em Baita *et al.* (1998), o problema pode ser classificado em:

- Topologia da rede de abastecimento: um para muitos;
- Itens a serem entregues: um;
- Demanda: determinística, comportamento constante ao longo do tempo e não-uniforme entre clientes;
- Decisões: instante de atendimento e quantidade;
- Restrições: capacidade dos veículos, capacidade de estocagem, capacidade de abastecimento, número de veículos fornecido e inexistência de demanda não atendida;
- Custos: custo de estoque proporcional ao volume estocado e custo de distribuição proporcional à distância;

- Estratégia de solução: meta-heurística populacional.

De acordo com a definição do problema em estudo, o custo de estoque é proporcional à quantidade estocada, tanto nos clientes quanto no fornecedor. Apesar de grande parte dos custos em problemas práticos serem fixos, como por exemplo, aluguel e depreciação das instalações, mão de obra, equipamentos de movimentação, alguns custos podem depender da quantidade estocada, como custos de perdas devido a roubo, obsolescência, avarias ou custo de seguro da quantidade estocada. No caso de caixas eletrônicos, por exemplo, o custo pode estar associado à quantidade de numerário que permanece disponível na máquina e sobre o qual não incide nenhum rendimento. Nesse caso, todos os clientes apresentam o mesmo custo. Outro aspecto a considerar é que diferentes clientes podem apresentar diferentes custos de estoque. Isso pode ocorrer, por exemplo, devido à localização do cliente ser ou não em uma área de maior risco.

Segundo a classificação em Ball (1998), o problema proposto segue um modelo dinâmico com tamanho de entrega variável, ou seja, a quantidade entregue é uma variável de decisão definida juntamente com o dia de atendimento.

É importante notar que, pelo fato do problema ser não periódico e de não existirem restrições sobre os níveis de estoque ao final do horizonte de planejamento, o problema admite soluções em que todos (ou quase todos) os clientes apresentam estoque zerado ao final do horizonte de planejamento. Isso implicará na necessidade de reabastecer todos os clientes no dia seguinte ao fim do horizonte de planejamento, o que muito provavelmente causará uma violação da capacidade de transporte. No entanto, para permitir a comparação com a solução mono-objetivo ótima existente na literatura, as mesmas restrições do problema original serão mantidas.

O problema foi escolhido por apresentar todas as características de um problema clássico de estoque e roteamento gerenciados pelo fornecedor. A existência, principalmente, das restrições de capacidade dos veículos e limite máximo de estoque, combinada com o uso do modelo de instante de atendimento, torna a busca por soluções bem desafiadora. Dada uma solução, qualquer alteração em uma alocação pode afetar tanto as alocações do mesmo cliente, em outros dias do horizonte de planejamento, quanto as alocações dos demais clientes atendidos no mesmo dia. Cada alocação afetada, por sua vez, tem influência em outras alocações, e assim por diante. Esse efeito em cascata torna o processo de melhoria das soluções muito complexo.

Este problema é considerado NP-completo, pois é possível reduzi-lo a um TSP (do inglês, *Traveling Salesman Problem*) ao se considerar um horizonte de planejamento com apenas um instante de tempo, inexistência de custos de estoque, capacidade do veículo infinita e que todos os clientes precisam ser atendidos.

Outra vantagem da formulação adotada é a existência de uma solução ótima para a sua formulação mono-objetivo, o que permite a comparação de resultados.

Para a solução do problema, foi adotada uma meta-heurística populacional por duas razões principais:

- O problema é NP-completo e, como os problemas reais dessa classe de problemas possuem, em geral, um espaço de busca grande, o uso de métodos exatos se torna proibitivo por questões de escalabilidade. Apesar dos softwares atuais que implementam métodos exatos serem extremamente eficientes, os problemas reais envolvem um elevado número de restrições e alta dimensão, gerando uma explosão combinatória de opções que inviabilizam o emprego de métodos exatos.
- O problema é inerentemente multiobjetivo. O uso de uma abordagem populacional torna possível disponibilizar rapidamente múltiplas soluções com diferentes compromissos entre os objetivos.

4.3 Formulação geral

Um produto é distribuído a partir de um fornecedor 0 a um conjunto $M=\{1,2,\dots,n\}$ de clientes ao longo de um horizonte de tempo H . A cada instante de tempo $t \in T=\{1,\dots,H\}$, uma quantidade r_0 é produzida e disponibilizada no fornecedor e a quantidade r_s é consumida no cliente $s \in M$. O fornecedor possui um nível de estoque inicial B_0 e cada cliente possui um nível de estoque inicial $I_{s,0}$. Cada cliente s admite um nível de estoque máximo dado por U_s . Se um cliente s é visitado no instante t , a quantidade de produto abastecido $x_{s,t}$ no cliente é tal que o nível de estoque não ultrapassa o valor máximo U_s .

Existem custos de estoque no fornecedor e nos clientes. O custo de estoque unitário no fornecedor é dado por h_0 e no cliente $s \in M$ é dado por h_s . Denominando B_t o nível de estoque do fornecedor no instante de tempo t , o custo de estoque total no fornecedor pode ser definido como:

$$\sum_{t \in T'} h_0 B_t \quad \text{onde } T' = T \cup \{H+1\}. \quad (4.1)$$

O instante de tempo $H+1$ é incluído para que seja possível levar em consideração as consequências dos reabastecimentos realizados no instante de tempo T . O custo total de estoque no cliente s é dado por:

$$\sum_{t \in T'} h_s I_{s,t} \quad \text{onde } T' = T \cup \{H+1\}, \quad (4.2)$$

sendo $I_{s,t}$ o nível de estoque do cliente s no instante t .

O problema é não-periódico, o que implica que os níveis de estoque ao final do horizonte de tempo podem diferir dos estoques iniciais.

O nível de estoque do fornecedor no instante $t \in T'$ é dado por:

$$B_t = B_{t-1} + r_0 - \sum_{s \in M} x_{s,t-1}. \quad (4.3)$$

De forma similar, define-se o nível de estoque do cliente $s \in M$ no instante $t \in T'$ como:

$$I_{s,t} = I_{s,t-1} + x_{s,t-1} - r_s. \quad (4.4)$$

O custo de transporte $c_{i,j}$ entre os clientes i e j é dado por:

$$c_{i,j} = NINT \left[\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right], \quad (4.5)$$

onde NINT é a função de arredondamento para o inteiro mais próximo e (x_i, y_j) são as coordenadas geográficas do cliente i .

O problema está sujeito às seguintes restrições:

1. Disponibilidade de estoque no fornecedor para o reabastecimento dos clientes:

$$B_t \geq \sum_{s \in M} x_{s,t}, \quad t \in T; \quad (4.6)$$

2. Disponibilidade de estoque no cliente:

$$I_{s,t} \geq 0 \quad s \in M, \quad t \in T'; \quad (4.7)$$

3. Limite de estoque máximo no cliente, fazendo com que a quantidade abastecida atenda à seguinte inequação:

$$x_{s,t} \leq U_s - I_{s,t} \quad s \in M, \quad t \in T; \quad (4.8)$$

4. Capacidade limitada de transporte:

$$\sum_{s \in M} x_{s,t} \leq C, \quad t \in T. \quad (4.9)$$

São dois os objetivos do problema:

1. Minimizar o custo total de estoque nos clientes e no fornecedor

$$C_{estoque} = \sum_{t \in T'} h_0 B_t + \sum_{s \in M} \sum_{t \in T'} h_s I_{s,t} \quad (4.10)$$

2. Minimizar o custo total de transporte

$$C_{transporte} = \sum_{i \in M'} \sum_{j \in M', j < i} \sum_{t \in T} c_{i,j} z_{i,j}^t \quad (4.11)$$

onde $z_{i,j}^t$ é uma variável binária que assume o valor 1 caso o cliente j suceda imediatamente o cliente i na rota do dia t , ou zero, caso contrário.

4.4 Formulação mono-objetivo \times multiobjetivo

Em Archetti *et al.* (2007), é usada uma formulação mono-objetivo cuja função-objetivo pondera os dois custos, atribuindo o mesmo peso a ambos. O objetivo é minimizar o custo total dado por:

$$\text{minimizar } C_{total} \equiv \text{minimizar } (C_{estoque} + C_{transporte}) \quad (4.12)$$

A proposta deste trabalho é resolver o problema usando uma formulação multiobjetivo, que busca minimizar os dois objetivos, dados pelas equações (4.10) e (4.11), explicitamente:

$$\text{minimizar } \begin{cases} C_{estoque} \\ C_{transporte} \end{cases} \quad (4.13)$$

A utilização de uma abordagem multiobjetivo permite que a escolha de uma solução que privilegie o transporte ou o estoque seja feita *a posteriori*. Além disso, a diferença de escala entre os custos

não afeta o desempenho da abordagem. Como poderemos ver na seção 4.6, metade das instância apresenta custo unitário de estoque no intervalo $(0,01; 0,05)$ e a outra metade no intervalo $(0,1; 0,5)$. Para as instâncias com custo de estoque no intervalo $(0,1; 0,5)$, o custo de transporte é equiparável em termos de escala. No caso dos custos de estoque no intervalo $(0,01; 0,05)$, o custo de transporte está numa escala acima da escala de custo de estoque. Dessa forma, usando a formulação mono-objetivo, a parcela do custo total relativo ao transporte predominaria sobre a do custo de estoque, podendo levar a resultados que privilegiam o custo de transporte.

4.5 Política de reabastecimento

Em Archetti *et al.* (2007), são apresentados resultados da solução exata para três políticas de reabastecimento.

- VMIR-OU ;
- VMIR-ML;
- VMIR.

As diferenças entre essas políticas baseiam-se na determinação da quantidade de produto a ser entregue ao cliente, sempre que ele é visitado. Na VMIR-OU, a quantidade é a máxima permitida pelo estoque corrente do cliente. Na política VMIR-ML, a quantidade pode ser qualquer valor que não ultrapasse o nível máximo de estoque do cliente. E na VMIR, a quantidade entregue pode ser qualquer valor, mesmo que o nível máximo de estoque do cliente seja ultrapassado. Nesse trabalho, optou-se por utilizar a política VMIR-ML por dois motivos:

- Em problemas reais, é comum a necessidade de se respeitar as restrições de estoque máximo, o que não ocorre com a política VMIR;
- Para a solução exata, a VMIR-OU mostrou resultados inferiores aos das demais, pois ela diminui a flexibilidade de alocação. Usando a política VMIR-OU, ao se definir o dia do reabastecimento, implicitamente se está definindo a quantidade a ser reabastecida. Nas políticas VMIR-ML e VMIR, o dia do reabastecimento e a quantidade a ser reabastecida são duas variáveis de decisão.

4.6 Instâncias de teste

Os testes foram realizados usando instâncias de *benchmark* encontradas na literatura. O objetivo desses testes foi validar a proposta em relação à convergência da solução. No entanto, como as

instâncias da literatura são de pequeno e médio porte, foram também geradas instâncias de grande porte e realizados testes para validação da escalabilidade da proposta.

4.6.1 Instâncias de *benchmark* da literatura

As instâncias de teste propostas em Archetti *et al.* (2007) foram geradas através de seleção aleatória de valores dentro de intervalos estabelecidos e seguindo uma distribuição uniforme. Estão divididas em 4 classes:

1. Horizonte = 3 dias; custo de estoque no fornecedor de 0,03 unidades; custo de estoque nos clientes: aleatoriamente gerado no intervalo (0,01; 0,05) (baixo custo de estoque);
2. Horizonte = 3 dias; custo de estoque no fornecedor de 0,3 unidades; custo de estoque nos clientes: aleatoriamente gerado no intervalo (0,1; 0,5) (alto custo de estoque);
3. Horizonte = 6 dias; custo de estoque no fornecedor de 0,03 unidades; custo de estoque nos clientes: aleatoriamente gerado no intervalo (0,01; 0,05) (baixo custo de estoque);
4. Horizonte = 6 dias; custo de estoque no fornecedor de 0,3 unidades; custo de estoque nos clientes: aleatoriamente gerado no intervalo (0,1; 0,5) (alto custo de estoque).

As instâncias de teste com horizonte de 3 dias compreendem casos com número de clientes no conjunto {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}. As de 6 dias compreendem casos com número de clientes no conjunto {5, 10, 15, 20, 25, 30}. Somam, no total, 160 instâncias de teste.

A Figura 4.6 exibe dados de uma instância de teste. No caso, a instância se enquadra na classe 4 e compreende 15 clientes.

16	6	1155						
1	249,0	454,0	1801	770	0,30			
2	449,0	228,0	11	22	0	11	0,20	
3	221,0	329,0	98	196	0	98	0,19	
4	64,0	54,0	170	255	0	85	0,24	
5	152,0	255,0	106	159	0	53	0,22	
6	128,0	498,0	94	188	0	94	0,40	
7	320,0	185,0	24	36	0	12	0,45	
8	306,0	330,0	42	84	0	42	0,16	
9	440,0	230,0	34	51	0	17	0,17	
10	151,0	453,0	59	118	0	59	0,26	
11	453,0	184,0	54	81	0	27	0,24	
12	67,0	211,0	62	124	0	62	0,41	
13	30,0	93,0	134	201	0	67	0,20	
14	453,0	90,0	10	20	0	10	0,25	
15	132,0	325,0	51	102	0	51	0,16	
16	194,0	380,0	82	164	0	82	0,18	

Fig. 4.6: Estrutura de dados de uma instância de teste

A primeira linha contém a seguinte informação:

n H C

onde:

n = número de clientes + 1 (conta também o fornecedor)

H = número de dias do horizonte de planejamento

C = capacidade máxima de transporte

A segunda linha contém informações do fornecedor:

1 x_1 y_1 B_1 r_1 h_1

onde:

1 = número correspondente ao fornecedor

x_1 = coordenada x do fornecedor

y_1 = coordenada y do fornecedor

$B1$ = estoque inicial no fornecedor

$r1$ = produção no fornecedor em cada dia do horizonte de planejamento

$h1$ = custo de estoque unitário no fornecedor

As linhas seguintes apresentam informações de cada cliente:

$s \quad x_s \quad y_s \quad Is0 \quad Us \quad Ls \quad rs \quad hs$

onde:

s = número do cliente

x_s = coordenada x do cliente s

y_s = coordenada y do cliente s

$Is0$ = nível de estoque inicial no cliente s

Us = nível máximo de estoque no cliente s

Ls = nível mínimo de estoque no cliente s (no problema considerado, esse valor é sempre zero)

rs = quantidade consumida no cliente s a cada dia do horizonte de planejamento

hs = custo de estoque unitário no cliente s

Para as 4 classes de instâncias de teste descritas, limitadas a 30 clientes para horizonte de planejamento de 6 dias, ou 50 clientes para horizonte de 3 dias, existe uma solução ótima para a formulação mono-objetivo do problema. A limitação no número de clientes decorre do fato do algoritmo exato só produzir resultado em tempo aceitável para problemas de dimensões reduzidas.

4.6.2 Proposta de novas instâncias

Duas classes adicionais de instância de teste são propostas nesse trabalho para que seja possível avaliar a abordagem de solução proposta em problemas de grande porte. Classe com horizonte de planejamento de 3 dias:

- Horizonte: 3 dias;
- Custo de estoque no fornecedor: 0,3 unidades;
- Custo de estoque nos clientes: aleatoriamente gerado no intervalo (0,1; 0,5);
- Número de clientes: 200

Classe com horizonte de planejamento de 6 dias:

- Horizonte: 6 dias;
- Custo de estoque no fornecedor: 0,3 unidades;
- Custo de estoque nos clientes: aleatoriamente gerado no intervalo (0,1; 0,5);
- Número de clientes: 200

Em problemas práticos com 200 clientes, provavelmente a frota de transporte não é limitada a um único veículo. No entanto, a proposta de se aplicar o algoritmo em problemas com 200 clientes e um único veículo de transporte visa exclusivamente verificar a escalabilidade do algoritmo.

4.7 Síntese do capítulo

Neste capítulo, foram apresentados em detalhes o IRP, a formulação matemática do problema, a política de reabastecimento e as instâncias de teste que foram usadas para a validação da abordagem proposta.

Capítulo 5

Estratégia de solução

A estratégia adotada para a solução do problema de estoque e roteamento gerenciados pelo fornecedor utiliza o algoritmo SPEA2 (ver seção 3.7). A escolha baseou-se no fato do SPEA2 ser considerado juntamente com NSGA-II, algoritmos evolutivos de referência na área de otimização multiobjetivo

O SPEA2, assim como outros algoritmos mencionados na seção 3.5, fornece apenas o mecanismo geral de busca. No entanto, para cada problema, é necessário definir a representação computacional das soluções-candidatas, a forma de geração da população inicial, o mapeamento do genótipo para o fenótipo dos indivíduos da população, os operadores de recombinação e mutação pertinentes ao problema, e o critério de dominância entre as soluções. Buscas locais não fazem parte do algoritmo SPEA2, no entanto, é reconhecido que a combinação de algoritmos evolutivos e buscas locais, além de tender a produzir soluções de melhor qualidade, pode acelerar o processo de convergência da solução. A abordagem proposta inclui buscas locais, combinadas ao SPEA2, que utilizam o critério de dominância para a avaliação das soluções.

As seções a seguir descrevem módulos do algoritmo que são específicos do problema tratado e, portanto, representam contribuições da pesquisa.

5.1 Critério de dominância

Ao invés do critério de dominância padrão, foi utilizado um algoritmo baseado no critério de dominância restrita (Deb (2001), Deb *et al.* (2002)), que permite a comparação entre duas soluções não-factíveis, ou seja, soluções que apresentam valor de violação maior que zero, assim como a comparação entre uma solução factível e outra não-factível.

Uma solução x apresenta uma dominância restrita sobre uma solução y se:

1. A solução x é factível e a solução y é infactível;

2. Ambas as soluções x e y são infactíveis, mas a solução x apresenta uma menor violação de restrições quando comparada à solução y ;
3. Ambas as soluções x e y são factíveis e x domina y .

Considerando:

- C_1 : soma dos custos totais de transporte de todas as rotas do horizonte de planeamento;
- C_2 : soma dos custos totais de estoque no fornecedor e nos clientes;
- V_1 : soma das violações de capacidade de transporte;
- V_2 : soma das violações de disponibilidade no fornecedor;
- V_3 : soma das violações de estoque mínimo do cliente;
- V_4 : soma das violações de estoque máximo do cliente;
- $I_c = \{1, 2\}$: índices dos custos;
- $I_v = \{1, \dots, 4\}$: índices das violações.

A existência de um valor de V_i ($i \in I_v$) maior que zero indica que a solução possui uma violação.

O Algoritmo 1 contém os passos para determinar se uma solução A domina uma solução B .

Algoritmo 1 Função de dominância

```

if  $(\forall i \in I_v, V_i(A) = 0)$  and  $(\exists i \in I_v \mid V_i(B) > 0)$  then {Se A não apresenta violações e B
apresenta pelo menos uma}
    return true; { A domina B }
else
    if  $(\exists i \in I_v \mid V_i(A) > 0)$  and  $(\forall i \in I_v, V_i(B) = 0)$  then {Se A apresenta pelo menos uma vio-
lação e B não apresenta nenhuma}
        return false; {A não domina B}
    else {A e B não apresenta nenhuma violação ou ambas apresentam violações}
        EstritamenteMenor = false;
        for  $i \in I_c$  do {Para todos os custos}
            if  $C_i(A) > C_i(B)$  then {Se A apresenta custo maior que B}
                return false; {A não domina B}
            else
                if  $C_i(A) < C_i(B)$  then {Se A apresenta pelo menos um custo menor que B}
                    EstritamenteMenor = true; {A pode dominar B se os outros custos forem menores
ou iguais aos de B}
                end if
            end if
        end for
        for  $i \in I_v$  do {Para todas as violações}
            if  $V_i(A) > V_i(B)$  then {Se A apresenta violação maior que B}
                return false; {A não domina B}
            else
                if  $V_i(A) < V_i(B)$  then {Se A apresenta pelo menos uma violação menor que B}
                    EstritamenteMenor = true; { A pode dominar B se as outras violações forem
menores ou iguais a B }
                end if
            end if
        end for
        return EstritamenteMenor; { Retorna true se A possui um custo ou violação menor que B
e demais custos e violações menores ou iguais}
    end if
end if

```

5.2 Codificação

O genótipo dos indivíduos da população é codificado em uma matriz $n \times m$, conforme apresentado na Figura 5.1 onde n é o número de clientes a serem atendidos e m é o tamanho do horizonte de planejamento (número de dias). O valor de cada elemento ij da matriz representa a quantidade de produto a ser entregue no cliente i no instante j . O valor 0 indica que não há reabastecimento. Note que, para cada dia t do horizonte de planejamento, é armazenado somente quais clientes são atendidos e o valor dos reabastecimentos, não havendo nenhuma informação da rota a ser seguida.

	D1	D2	D3	D4	D5	D6
C1	0	130	0	0	195	65
C2	91	0	0	273	0	0
C3	75	0	150	75	0	75
C4	0	86	0	0	129	43
C5	64	0	128	64	0	128

Fig. 5.1: Codificação do genótipo

Na geração da população inicial, são criadas matrizes binárias com 0's e 1's atribuídos aleatoriamente. O valor 1 na coluna ij indica que deve ocorrer um abastecimento do cliente i no instante j . A transformação dos 1's em valores de alocação é feita a partir do primeiro dia do horizonte de alocação. Para todos os clientes do primeiro dia, é calculado o estoque após o consumo diário. Para todos os clientes que apresentarem estoque negativo ou tiverem um 1 na matriz binária, é calculado o abastecimento mínimo para suprir o consumo até o próximo reabastecimento. O valor de reabastecimento é limitado pelo estoque máximo suportado no cliente. O procedimento é aplicado para os dias seguintes do horizonte de planejamento. Esse processo garante que não haverá violação de estoque mínimo/máximo nos clientes. No entanto, a solução pode apresentar violação de capacidade máxima de transporte e de produção do fornecedor. Caso haja violações, são feitos rearranjos visando minimizar ou eliminar as violações. Os rearranjos são executados em dois passos. No primeiro passo, para cada dia com violação de capacidade de transporte, tenta-se realocar a quantidade excedida para o reabastecimento anterior ou posterior. Caso ainda haja violações, é executado o segundo passo, onde são criadas novas alocações (em dias anteriores ou posteriores que apresentem capacidade disponível) e as quantidades excedidas são movidas para as nova alocações. Não há garantia de que a solução gerada seja factível, mas o algoritmo foi estruturado para trabalhar com soluções factíveis e não-factíveis, conforme descrito na seção 5.1. O pseudocódigo do cálculo de alocações está descrito no Algoritmo 2.

Em Archetti *et al.* (2011), é proposta uma abordagem de otimização baseada em busca tabu. Como a proposta trabalha com apenas uma solução corrente, entre uma iteração e outra do algoritmo, são armazenadas tanto as quantidades de reabastecimento quanto as rotas a serem seguidas. Na codificação proposta nesse trabalho, optou-se por usar uma codificação em forma de matriz que armazena as quantidades de reabastecimento. As rotas a serem seguidas são calculadas sempre que necessário, empregando o software Concorde (ver seção 5.6). A decisão foi tomada baseada nas seguintes análises:

- A codificação em matriz binária é extensamente utilizada em algoritmos evolutivos. A utilização de matrizes torna a aplicação de operadores de recombinação e mutação mais diretas. Apesar da sua forma mais usual ser em matriz binária, optou-se por manter na matriz, além da informação de quando o cliente é abastecido, o valor do reabastecimento. O uso de uma matriz binária exigiria a implementação de um algoritmo determinístico e muito eficiente de cálculo das quantidades de abastecimento. Em contrapartida, o uso de uma matriz com as quantidades de reabastecimento permite que a definição das quantidades de reabastecimento sejam o produto do próprio algoritmo evolutivo e das buscas locais empregadas.
- Para o cálculo da melhor rota, foi escolhida uma implementação muito eficiente da heurística de Chained-Lin-Kernighan, que resolve problemas de TSP (do inglês, *Travelling Salesman Problem*). Não há contribuições da pesquisa neste módulo de otimização da rota. Ou seja, é relativamente rápido obter as rotas a partir do conjunto de clientes a serem visitados, presente no software Concorde (Concorde Home Page (n.d.)).
- O armazenamento das rotas para todos os dias do horizonte de planejamento de todos os indivíduos da população exigiria uma estrutura mais complexa, que consumiria muito espaço em memória.
- A inclusão de cidades a mais em um problema de TSP pode mudar completamente a solução ótima do problema. Como a abordagem proposta utiliza extensivamente operadores de mutação e recombinação, é muito comum que o conjunto de clientes a serem visitados seja alterado. A determinação de quando seria possível reaproveitar ou não uma rota existente exigiria um esforço computacional similar ao de recalculá-las sempre que necessário.

5.3 Operadores de recombinação

Em algoritmos evolutivos, os mecanismos de busca são inspirados nos conceitos da Teoria de Evolução das Espécies. Esses mecanismos apresentam um uso intensivo de operadores de recombi-

Algoritmo 2 Cálculo de alocação

```

for  $t$  em {horizonte de programação} do
  for  $c$  em {conjunto de clientes} do
    Calcula estoque atual
    if estoque atual  $< 0$  or matriz binária( $t, c$ ) == 1 then
      Calcula reabastecimento necessário para suprir demanda até próxima alocação definida na
      matriz binária e atualiza matriz de alocação
    end if
  end for
end for
for  $i$  em {tentativa1, tentativa2} do
  for  $t$  em {horizonte de programação} do
    Verifica violação de transporte e/ou produção do fornecedor
    if ocorreu violação then
      for  $c$  in {conjunto de clientes com abastecimento em  $t$ } do
        if  $i == tentativa1$  then
          if violação persiste then
            Obtém o dia  $t_{ant}$  imediatamente anterior a  $t$  em que o cliente  $c$  possua um
            reabastecimento.
            if existe capacidade de estoque, de produção e de transporte no instante  $t_{ant}$  then
              Move alocação do instante  $t$  para  $t_{ant}$  de acordo com a capacidade
            end if
            if violação persiste then
              Obtém o dia  $t_{post}$  imediatamente posterior a  $t$  em que o cliente  $c$  possua um rea-
              bastecimento.
              if existe capacidade de estoque, de produção e de transporte no instante  $t_{post}$  then
                Move alocação do instante  $t$  para  $t_{post}$  de acordo com a capacidade
              end if
            end if
          end if
        else { $i == tentativa2$ }
          Obtém o dia  $t_{ant}$  imediatamente anterior a  $t$  e que tenha capacidade de estoque, de
          produção e de transporte
          Cria novo reabastecimento para o cliente  $c$  em  $t_{ant}$  e move alocação de  $t$  para  $t_{ant}$ 
          if violação persiste then
            Obtém o dia  $t_{post}$  imediatamente posterior a  $t$  e que tenha capacidade de estoque, de
            produção e de transporte
            Cria novo reabastecimento para o cliente  $c$  em  $t_{post}$  e move alocação de  $t$  para  $t_{post}$ 
          end if
        end if
      end for
    end if
  end for
end if
end for

```

nação (do inglês *crossover*). A recombinação é o mecanismo pelo qual novas soluções são geradas a partir da combinação entre soluções existentes. Na medida em que os indivíduos são selecionados de acordo com a respectiva qualidade, o uso de operadores de recombinação tende a aumentar mais rapidamente a qualidade média da população, se as partes boas de cada solução combinada forem preservadas. A recombinação envolve dois indivíduos da população P_1 e P_2 e gera dois novos indivíduos F_1 e F_2 . Parte do material genético de P_1 e parte do material genético de P_2 são combinados para gerar o F_1 . As partes não utilizadas para gerar F_1 são empregadas na geração do indivíduo F_2 . Após a recombinação, são executadas heurísticas simples para tentar factibilizar as soluções geradas. Caso as violações permaneçam, as quantidades de reabastecimento são recalculadas para toda a matriz. Isso não garante que a solução gerada seja sempre factível, mas aumenta consideravelmente a possibilidade disso ocorrer.

Foram implementados os seguintes operadores de recombinação:

- Recombinação vertical de um ponto;
- Recombinação horizontal de um ponto;
- Recombinação uniforme de células;
- Recombinação uniforme de colunas;
- Recombinação uniforme de linhas.

Os operadores de recombinação são executados em 60% da população. Cada um dos cinco tipos de recombinação tem igual probabilidade de ocorrência. A probabilidade de ocorrência de recombinação, 60%, foi definida empiricamente por tentativa e erro.

5.3.1 Recombinação vertical de um ponto

É feito um corte vertical em uma coluna escolhida aleatoriamente nos dois indivíduos a serem recombinados. Cada descendente herda uma parte do genótipo de cada progenitor.

No exemplo apresentado na Figura 5.2, o corte ocorre na coluna D4. Um dos descendentes herda as colunas D1-D4 de um progenitor e as colunas D5-D6 do outro. O outro descendente herda o complemento.

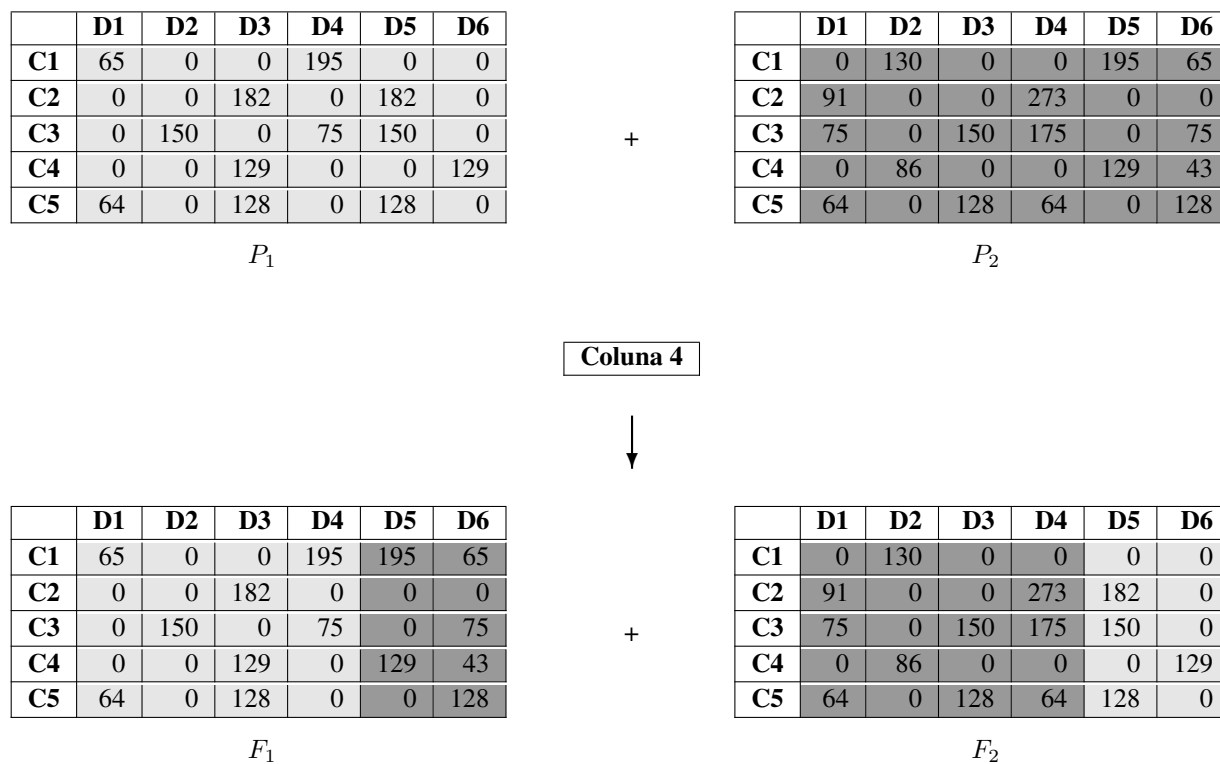


Fig. 5.2: Recombinação vertical de um ponto

Motivação intuitiva: Políticas de abastecimento diário (colunas inteiras) presentes nos indivíduos P_1 e P_2 são combinadas nos indivíduos F_1 e F_2 . Caso os indivíduos P_1 e P_2 possuam boas políticas apenas para alguns dias consecutivos, cria-se a possibilidade de que ao menos um dos indivíduos F_1 e F_2 passe a conter as boas políticas de abastecimento diárias de P_1 e P_2 .

5.3.2 Recombinação horizontal de um ponto

É feito um corte horizontal em uma linha escolhida aleatoriamente nos dois indivíduos a serem recombinados. Cada descendente herda uma parte do genótipo de cada progenitor.

No exemplo apresentado na Figura 5.3, o corte ocorre na linha C1. Um dos descendentes herda a linha C1 de um progenitor e as linhas C2-C6 do outro. O outro descendente herda o complemento.

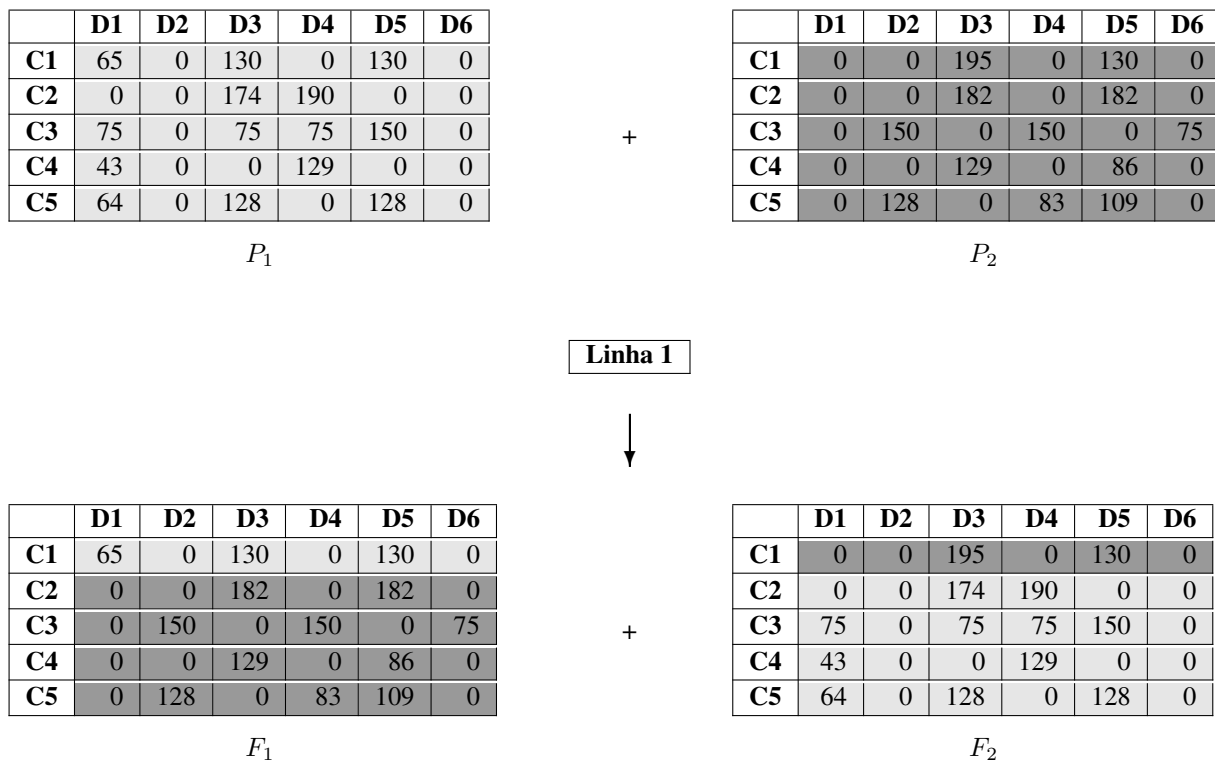


Fig. 5.3: Recombinação horizontal de um ponto

Motivação intuitiva: Políticas de abastecimento de clientes, durante todo o horizonte de planejamento (linhas inteiras), presentes nos indivíduos P_1 e P_2 são combinados nos indivíduos F_1 e F_2 . Caso os indivíduos P_1 e P_2 possuam boas políticas em apenas um subconjunto de clientes, cria-se a possibilidade de que ao menos um dos indivíduos F_1 e F_2 passe a conter as boas políticas de abastecimento de clientes presentes em P_1 e P_2 .

5.3.3 Recombinação uniforme de células

É gerada aleatoriamente um mapa de bits com a mesma dimensão da matriz que representa o genótipo dos indivíduos. Um dos descendentes herda as células de um dos progenitores especificadas como 1 no mapa de bits e as células de outro progenitor especificadas como 0 no mapa de bits. O outro descendente herda o complemento. O termo "célula" está sendo usado nesse contexto como sinônimo de "elemento ij " da matriz.

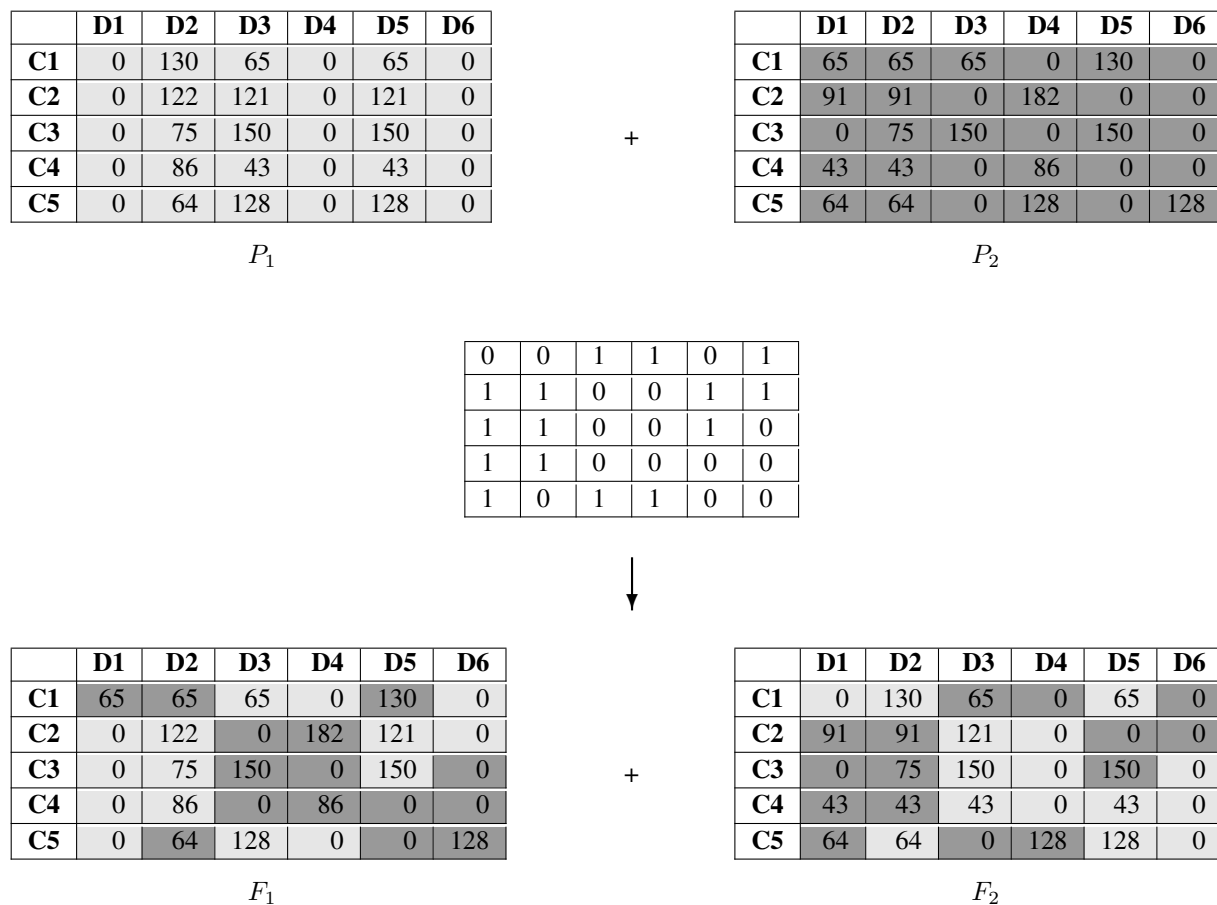


Fig. 5.4: Recombinação uniforme de células

Motivação intuitiva: É indicado no caso de P_1 e P_2 não disporem nem de boas políticas de abastecimento diário (coluna toda), nem de boas políticas de abastecimento de clientes durante o horizonte de planejamento (linha toda), mas apresentarem boas decisões de abastecimento isoladas e que podem ser combinadas em F_1 e F_2 .

5.3.4 Recombinação uniforme de colunas

É gerada aleatoriamente uma sequência de bits formando um vetor de tamanho igual ao número de colunas da matriz que representa o genótipo dos indivíduos. Cada um dos descendentes recebe as colunas de um dos progenitores, especificadas como 1 no vetor de bits, e as colunas de outro progenitor, especificadas como 0 no vetor de bits. No exemplo, um dos descendentes herda as colunas [D1,D2,D3,D5] de um dos progenitores e [D4,D6] do outro progenitor. O outro descendente herda o complemento.

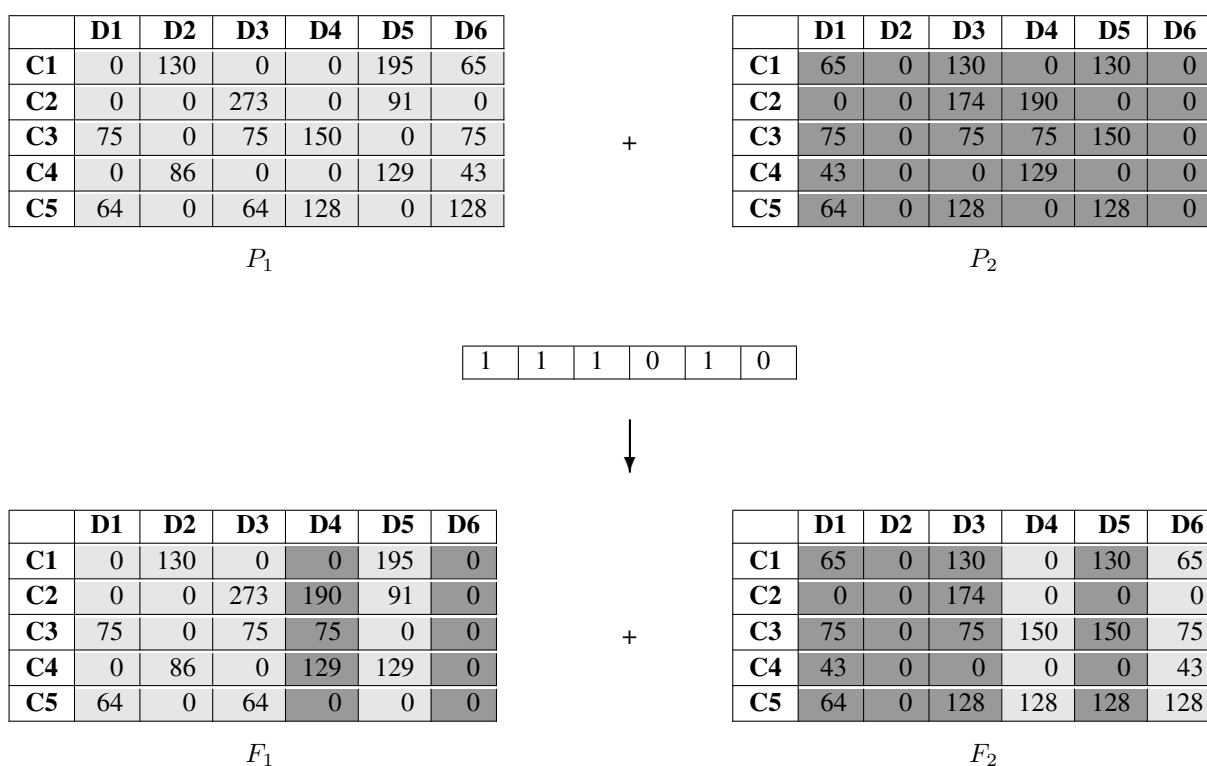


Fig. 5.5: Recombinação uniforme de colunas

Motivação intuitiva: Similar ao caso do operador de recombinação vertical de um ponto (ver seção 5.3.1), mas aqui as colunas a serem tomadas de P_1 e P_2 não precisam ser consecutivas.

5.3.5 Recombinação uniforme de linhas

É gerada aleatoriamente uma sequência de bits formando um vetor de tamanho igual ao número de linhas da matriz que representa o genótipo dos indivíduos. Cada um dos descendentes recebe as linhas de um dos progenitores, especificadas como 1 no vetor de bits, e as linhas de outro progenitor, especificadas como 0 no vetor de bits. No exemplo, um dos descendentes herda as linhas [C1,C3,C5] de um dos progenitores e [C2,C4] do outro progenitor. O outro descendente herda o complemento.

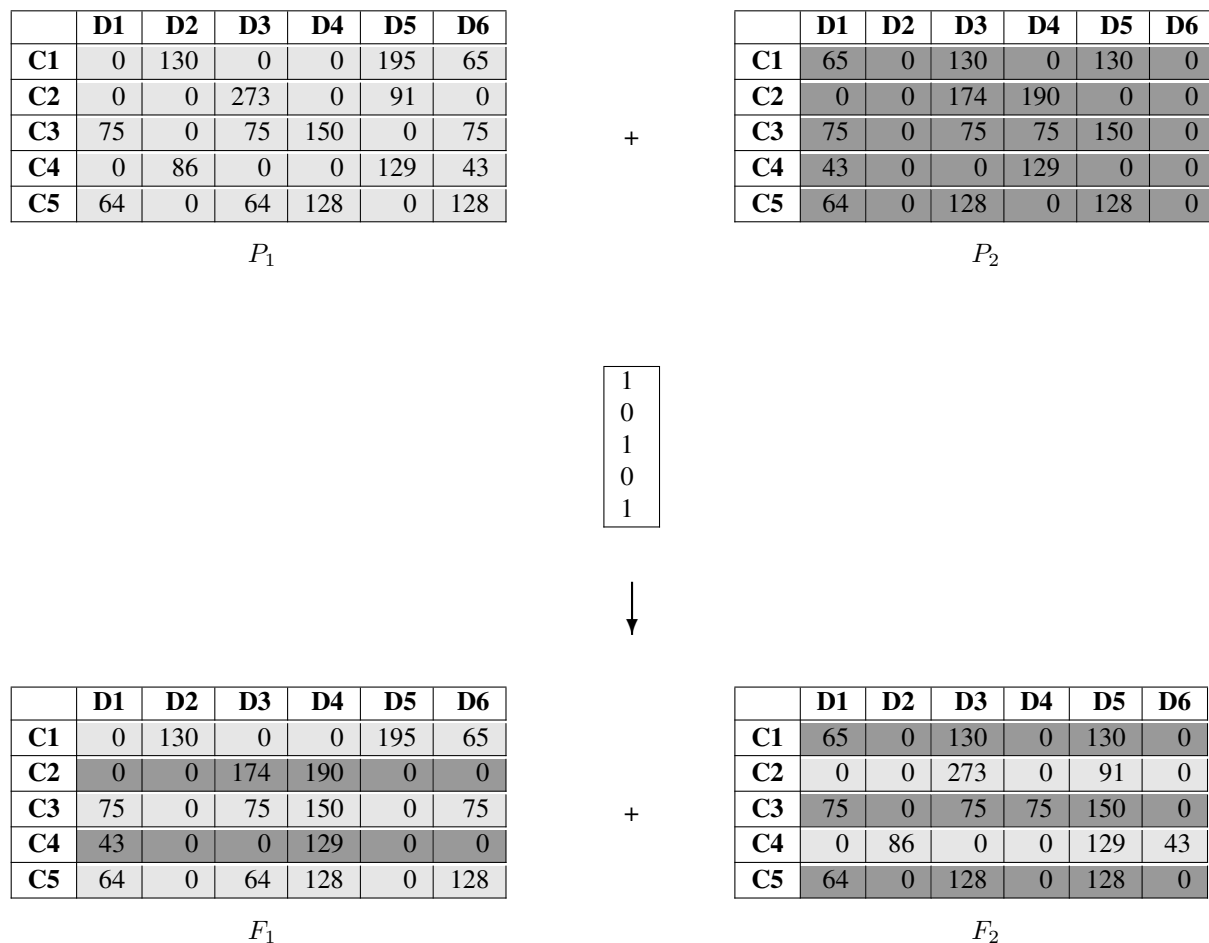


Fig. 5.6: Recombinação uniforme de linhas

Motivação intuitiva: Similar ao caso do operador de recombinação horizontal de um ponto (ver seção 5.3.2), mas aqui as linhas a serem tomadas de P_1 e P_2 não precisam ser consecutivas.

5.4 Operadores de mutação

O operador de mutação visa facilitar a exploração de diferentes partes do espaço de busca. A mutação cria um indivíduo modificado que é copiado para a próxima geração da população. A mutação consiste na promoção de uma mudança aleatória no material genético de um indivíduo. Após a mutação, são executadas heurísticas simples para tentar factibilizar as soluções geradas. Caso as violações permaneçam, as quantidades de reabastecimento são recalculadas para toda a matriz.

Foram implementados os seguintes operadores de mutação:

- União de reabastecimentos;
- Inserção de reabastecimento;
- Troca de reabastecimentos.

Apesar do operador de mutação ser, em geral, aplicado a uma população com uma probabilidade baixa de ocorrência, esse operador foi utilizado na abordagem implementada com uma probabilidade alta de ocorrência, de 80%. Este valor foi definido de forma empírica, por tentativa e erro. Como o algoritmo SPEA2 é elitista, já se garante que as N melhores soluções serão sempre preservadas (N é um parâmetro do algoritmo que define o tamanho do arquivo, ver seção 3.7). O uso intensivo de operadores de mutação possibilitou uma exploração melhor do espaço de busca, gerando melhores resultados.

5.4.1 União de reabastecimentos

Duas colunas, escolhidas aleatoriamente, são combinadas em uma única coluna. A escolha de qual coluna é zerada e qual coluna recebe a combinação das colunas é feita aleatoriamente.

No exemplo, as colunas D2 e D5 são combinadas e atribuídas à coluna D5. A coluna D2 é zerada.

	D1	D2	D3	D4	D5	D6
C1	0	65	0	195	0	0
C2	182	0	182	0	0	0
C3	75	0	75	75	75	75
C4	0	0	129	0	0	129
C5	128	128	64	0	0	0

$$D5 = D2 + D5$$



	D1	D2	D3	D4	D5	D6
C1	0	0	0	195	65	0
C2	182	0	182	0	0	0
C3	75	0	75	75	75	75
C4	0	0	129	0	0	129
C5	128	0	64	0	128	10

Fig. 5.7: União de reabastecimentos

Motivação intuitiva: Reduzir o número de rotas, visando baixar o custo de transporte. Supõe-se que a capacidade de abastecimento atende com alguma folga a demanda.

5.4.2 Inserção de reabastecimento

São escolhidas aleatoriamente uma linha e uma coluna para a inserção de um reabastecimento. É recalculado o reabastecimento do cliente na linha seleccionada usando o Algoritmo 2 para apenas um cliente.

No exemplo, é inserido um reabastecimento na célula C3/D3.

	D1	D2	D3	D4	D5	D6
C1	0	0	195	0	0	0
C2	0	0	182	0	0	0
C3	0	150	0	150	0	75
C4	0	0	129	0	86	0
C5	0	128	0	83	109	128

C3 / D3



	D1	D2	D3	D4	D5	D6
C1	0	0	195	0	0	0
C2	0	0	182	0	0	0
C3	0	150	75	75	0	75
C4	0	0	129	0	86	0
C5	0	128	0	83	109	128

Fig. 5.8: Inserção de reabastecimento

Motivação intuitiva: Toda inclusão de abastecimento tende a aumentar o custo de transporte, mas em contrapartida tende a reduzir o custo de estoque, caso realizada em cliente com custo de estoque mais baixo que o fornecedor, após recalculer o reabastecimento na respectiva linha.

5.4.3 Troca de reabastecimentos

São escolhidas aleatoriamente duas colunas e é feita uma troca dos reabastecimentos das duas colunas.

No exemplo, ocorre a troca das colunas D3 e D6. Este operador é semelhante a um dos procedimentos descrito em Archetti *et al.* (2011) como parte de uma fase de melhoria da solução utilizando MIP (do inglês *Mixed-Integer Programming*).

	D1	D2	D3	D4	D5	D6
C1	0	0	195	0	130	0
C2	0	0	174	190	0	0
C3	75	0	75	75	150	0
C4	43	0	0	129	0	0
C5	64	0	128	0	128	0

D3 <-> D6



	D1	D2	D3	D4	D5	D6
C1	0	0	0	0	130	195
C2	0	0	0	190	0	174
C3	75	0	0	75	150	75
C4	43	0	0	129	0	0
C5	64	0	0	0	128	128

Fig. 5.9: Troca de reabastecimentos

Motivação intuitiva: Produz melhorias caso existam dias com boas soluções de rota, mas que podem reduzir os custo de estoque ou gerar menos violações quando atribuídas a outros dias.

5.5 Operadores de busca local

A busca local corresponde a um procedimento que permite uma busca sistemática no espaço de soluções-candidatas, por meio de alterações que permitam encontrar soluções vizinhas. As soluções obtidas através da busca local são somente empregadas caso haja uma melhoria da solução. Em otimização multiobjetivo, uma melhoria na solução implica em encontrar uma solução que domine

a solução existente. O uso de busca local em combinação com algoritmos genéticos são conhecidos como algoritmos genéticos híbridos ou algoritmos meméticos (ver Moscato (1989)).

Foram implementados os seguintes operadores de busca local:

- Remoção de reabastecimentos;
- Deslocamento de reabastecimento;
- Troca de reabastecimento em rota de alto custo;
- Alocação máxima em rota de baixo custo de estoque;
- União e separação de rotas.

As quatro primeiras buscas locais são executadas para toda a população, imediatamente antes do processo "Atribuição de Fitness" (ver Figura 3.7). A busca local de união e separação de rotas exige um custo computacional maior e é executada apenas para 10% das soluções que apresentam melhor valor da função de *fitness*. Essa busca é executada ao fim do processo "Seleção de Pares" (ver Figura 3.7).

5.5.1 Remoção de reabastecimentos

É executada a remoção de reabastecimento em um cliente c em um dia t . A carga desse dia é distribuída nos dias anteriores e/ou posteriores a t , dias estes em que há reabastecimento. Este operador é executado para todos os reabastecimentos planejados, sempre que o estoque no cliente e dia considerados ultrapassar a demanda diária.

No exemplo, o reabastecimento na célula C2/D2 é removido. O reabastecimento que ocorreria nesse dia é movido para outro instante de tempo em que já está planejado um reabastecimento. No caso, o da célula C2/D3.

	D1	D2	D3	D4	D5	D6
C1	0	130	65	65	0	0
C2	0	91	91	182	0	0
C3	0	75	75	150	0	75
C4	0	83	43	46	0	0
C5	0	128	64	64	0	128

C2 / D2



	D1	D2	D3	D4	D5	D6
C1	0	130	65	65	0	0
C2	0	0	182	182	0	0
C3	0	75	75	150	0	75
C4	0	83	43	46	0	0
C5	0	128	64	64	0	128

Fig. 5.10: Remoção de reabastecimentos

Motivação intuitiva: Reduzir custo de transporte através da diminuição do número de clientes visitados nas rotas.

5.5.2 Deslocamento de reabastecimento

É executado um deslocamento de reabastecimento para o instante de tempo seguinte ou anterior. O horizonte de planejamento é considerado de forma circular, ou seja, o instante anterior ao primeiro dia do horizonte corresponde ao último dia do horizonte. Da mesma forma, o instante seguinte ao último dia do horizonte de planejamento corresponde ao primeiro dia. Este operador é executado para todos os reabastecimentos planejados.

No exemplo, o reabastecimento que ocorreria no cliente C2 no instante D1 é deslocado para o instante D6.

	D1	D2	D3	D4	D5	D6
C1	0	0	195	65	0	0
C2	91	0	91	182	0	0
C3	0	150	0	149	0	75
C4	43	0	82	47	0	0
C5	0	128	64	64	0	128

C2: D1 <-> D6



	D1	D2	D3	D4	D5	D6
C1	0	0	195	65	0	0
C2	0	0	91	182	0	91
C3	0	150	0	149	0	75
C4	43	0	82	47	0	0
C5	0	128	64	64	0	128

Fig. 5.11: Deslocamento de reabastecimento

Motivação intuitiva: Este operador pode resultar em diminuição de custo de estoque caso haja deslocamento de um reabastecimento para o dia anterior, em um cliente com custo de estoque menor que o do fornecedor, ou um deslocamento de reabastecimento para o dia seguinte, em um cliente com custo de estoque maior que o do fornecedor. Pode ocorrer também, uma diminuição no custo de transporte, caso o reabastecimento seja deslocado para um dia em que a rota contenha vizinhos mais próximos ao cliente cujo reabastecimento está sendo deslocado.

5.5.3 Troca de reabastecimento em rota de alto custo

É realizada a troca de reabastecimento de cliente em rota com alto custo para rota com vizinhos mais próximos. Escolhe-se a rota com maior custo, ou seja, a que apresenta o maior valor para o índice $\frac{custo_{rota}}{|rota|}$, onde $|rota|$ é o número de clientes abastecidos na rota. Remove-se da rota o reabastecimento do cliente mais distante dos demais membros da rota. Em seguida, procura-se a rota onde o cliente removido possua vizinhos mais próximos para se incluir o reabastecimento. Este operador é executado uma vez para cada solução-candidata.

No exemplo, a rota que apresenta maior custo é a do dia D1, sendo que o cliente mais distante dos demais clientes da rota e/ou do depósito é o cliente C2. Este reabastecimento é movido para o dia D2, cuja rota inclui clientes mais próximos do cliente C2. Essa troca causou um recálculo de reabastecimento do cliente C2, como pode ser observado nos dias D3, D5 e D6.

	D1	D2	D3	D4	D5	D6
C1	0	0	195	0	0	170
C2	91	0	182	0	0	91
C3	0	150	0	75	75	75
C4	43	0	86	0	0	43
C5	0	128	0	128	0	128

D1 / C2



	D1	D2	D3	D4	D5	D6
C1	0	0	195	0	0	170
C2	0	182	0	0	182	0
C3	0	150	0	75	75	75
C4	43	0	86	0	0	43
C5	0	128	0	128	0	128

Fig. 5.12: Troca de reabastecimento em rota de alto custo

Motivação intuitiva: O objetivo é reduzir o custo de transporte ao mover um reabastecimento pertencente a uma rota de alto custo para uma rota com vizinhos mais próximos.

5.5.4 Alocação máxima em cliente de baixo custo de estoque

Selecionam-se os clientes com custo de estoque menor que o fornecedor. Em seguida ordena-se esses clientes baseado no custo de estoque e aumenta-se o reabastecimento nesses clientes até o limite da capacidade do cliente e de transporte, de acordo com a ordem estabelecida previamente, ou seja, privilegiando os clientes com menor custo de estoque. São considerados apenas os casos em que já ocorre reabastecimento.

No exemplo, os clientes C1, C4 e C5 apresentam custo de estoque mais baixo que o fornecedor, e recebem alocação máxima possível, respeitando-se seus limites máximos de estoque e a capacidade de transporte no dia.

	D1	D2	D3	D4	D5	D6
C1	0	130	65	0	0	65
C2	0	0	182	0	182	0
C3	0	75	150	0	75	75
C4	0	83	46	0	0	43
C5	0	128	64	0	64	64

C1 / C4 / C5



	D1	D2	D3	D4	D5	D6
C1	0	130	65	0	0	195
C2	0	0	182	0	182	0
C3	0	75	150	0	75	75
C4	0	83	46	0	0	129
C5	0	128	64	0	128	164

Fig. 5.13: Alocação máxima em clientes de baixo custo de estoque

Motivação intuitiva: O objetivo é reduzir o custo total de estoque aumentando as quantidades de reabastecimentos já existentes até o limite do estoque do cliente, para todos os clientes com custo de estoque menor que o fornecedor.

5.5.5 União e separação de rotas I

Unem-se os reabastecimentos de dois instantes de tempo e os separa em duas rotas de menor custo no total. A separação é executada de forma a dividir a rota em duas partes tal que elas fiquem em posições opostas em relação ao depósito.

No exemplo, as rotas dos dias D5 e D6, escolhidas aleatoriamente, são unidas e melhor divididas em duas rotas que apresentam vizinhos mais próximos. A Figura 5.15 mostra as rotas antes e depois da aplicação do operador.

	D1	D2	D3	D4	D5	D6
C1	0	0	117	0	39	0
C2	0	0	95	285	0	0
C3	0	0	39	0	13	0
C4	110	0	0	55	110	0
C5	0	141	0	0	47	0
C6	78	0	0	39	78	0
C7	178	0	89	178	0	0
C8	178	0	89	178	10	0
C9	0	185	0	0	123	0
C10	0	192	0	96	192	0

D2 + D3 -> D2 + D3



	D1	D2	D3	D4	D5	D6
C1	0	117	0	0	39	0
C2	0	0	95	285	0	0
C3	0	39	0	0	13	0
C4	110	0	0	55	110	0
C5	0	141	0	0	47	0
C6	78	0	0	39	78	0
C7	178	0	89	178	0	0
C8	178	0	89	178	10	0
C9	0	0	185	0	123	0
C10	0	0	192	96	192	0

Fig. 5.14: União e separação de rotas I

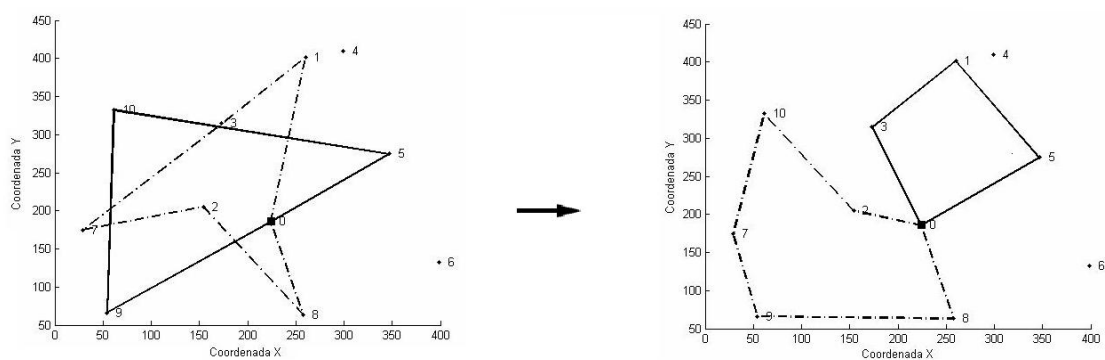


Fig. 5.15: Rotas antes e depois da aplicação do operador

Motivação intuitiva: Reduzir o custo de transporte através de uma melhor combinação de rotas.

5.5.6 União e separação de rotas II

Similar ao operador descrito na seção 5.5.5. A diferença é que as rotas são separadas de forma que se forme uma rota com os clientes mais próximos do depósito e outra rota com os clientes mais distantes do depósito. A Figura 5.17 mostra as rotas antes e depois da aplicação do operador.

	D1	D2	D3	D4	D5	D6
C1	0	0	117	0	39	0
C2	0	0	95	285	0	0
C3	0	0	39	0	13	0
C4	110	0	0	55	110	0
C5	0	141	0	0	47	0
C6	78	0	0	39	78	0
C7	178	0	89	178	0	0
C8	178	0	89	178	10	0
C9	0	185	0	0	123	0
C10	0	192	0	96	192	0

D2 + D3 -> D2 + D3



	D1	D2	D3	D4	D5	D6
C1	0	0	117	0	39	0
C2	0	95	0	285	0	0
C3	0	39	0	0	13	0
C4	110	0	0	55	110	0
C5	0	0	141	0	47	0
C6	78	0	0	39	78	0
C7	178	0	89	178	0	0
C8	178	0	89	178	10	0
C9	0	0	185	0	123	0
C10	0	0	192	96	192	0

Fig. 5.16: União e separação de rotas II

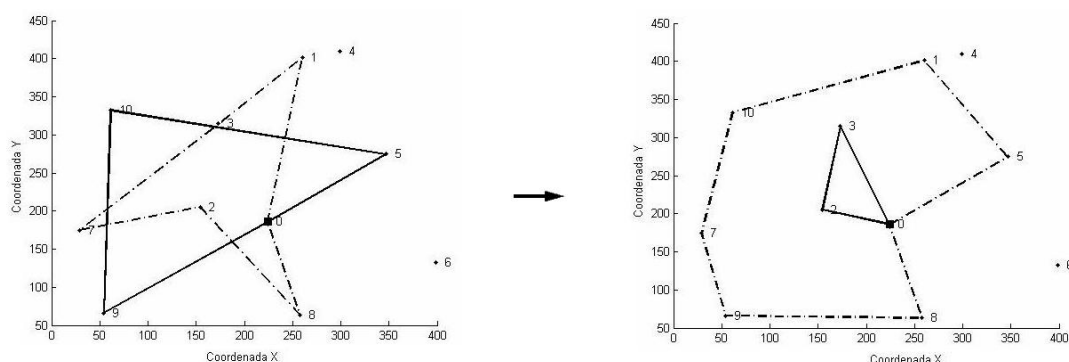


Fig. 5.17: Rotas antes e depois da aplicação do operador

5.6 Implementação

A implementação do algoritmo foi feita em Matlab. A Figura 5.18 mostra uma decomposição hierárquica do algoritmo SPEA2 em módulos. No módulo de cálculo do *fitness* bruto, é feito o cálculo do custo total de transporte, do custo total de estoque e dos valores das violações. Para o cálculo do custo de transporte, primeiro é resolvido um TSP (do inglês *Traveling Salesman Problem*) para cada dia do horizonte de planejamento. Em seguida, é feito o cálculo do custo de transporte de acordo com a melhor rota encontrada. O TSP é resolvido usando uma implementação da heurística de Chained-Lin-Kernighan denominada Linkern e disponível no site do projeto Concorde (<http://www.tsp.gatech.edu/concorde>). Foi implementado também um módulo de integração entre o Matlab e a biblioteca do Linkern. O módulo de integração foi implementado na linguagem C.

5.7 Síntese do capítulo

Neste capítulo, foi apresentada uma abordagem evolutiva multiobjetivo baseada no algoritmo SPEA2. Foram apresentados em detalhes o critério de dominância, a codificação dos indivíduos, os operadores de mutação, recombinação e busca local concebidos na abordagem, assim como a motivação para o uso de cada um. É apresentada também uma visão geral da implementação da abordagem proposta.

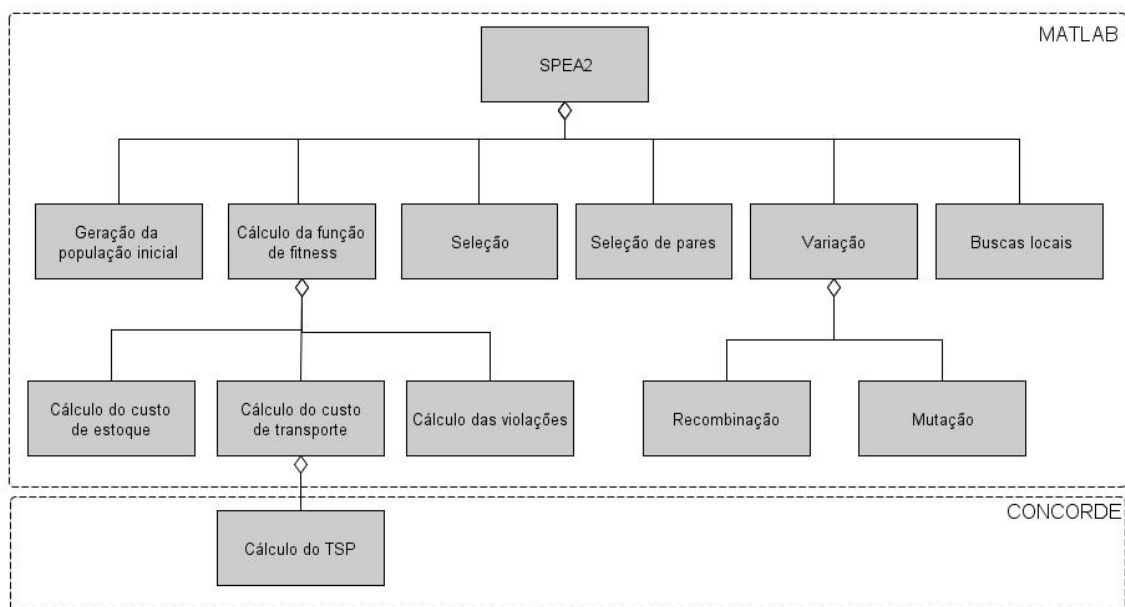


Fig. 5.18: Módulos do algoritmo empregado para otimização multiobjetivo

Capítulo 6

Resultados

O objetivo deste capítulo é avaliar a qualidade das soluções obtidas para o problema de otimização multiobjetivo de estoque e roteamento gerenciados pelo fornecedor, de acordo com dois critérios:

- Convergência das soluções;
- Diversidade das soluções.

Em relação à nomenclatura, será utilizado o termo ‘conjunto de Pareto’ para as soluções não-dominadas obtidas pela abordagem proposta. O termo ‘conjunto Pareto-ótimo’ será usado para se referir à fronteira de Pareto de cada problema, ou seja, o conjunto ideal de soluções que, por sinal, não é conhecido. E ‘solução mono-objetivo’ será usado para referenciar a solução ótima da formulação mono-objetivo disponível na literatura, a qual simplesmente soma o valor das duas funções-objetivo.

A abordagem de solução multiobjetivo é inédita para o problema apresentado, o que significa que não existe na literatura o conjunto Pareto-ótimo para as instâncias do problema e nem mesmo outros conjuntos de Pareto para comparação. Isso impossibilita a utilização de várias métricas comumente usadas na análise de desempenho dos algoritmos multiobjetivos e descritas na seção 3.6. As métricas que podem ser empregadas diretamente são a de espaçamento (equação 3.9), número de nichos (equação 3.12) e hipervolume (equação 3.15), que representam métricas de avaliação de diversidade (o hipervolume também é uma métrica de convergência). No presente trabalho, foi usada a métrica de espaçamento.

Em relação à convergência da solução, como está disponível na literatura somente a solução ótima para a formulação mono-objetivo do problema, a avaliação foi feita comparando-se a solução mono-objetivo com a solução do conjunto de Pareto obtido que mais se aproxima da solução mono-objetivo no espaço dos objetivos. Essa avaliação representa uma adaptação da métrica de distância geracional dada pela equação (3.6), para o caso em que só existe uma solução a ser comparada.

6.1 Apresentação de uma solução do conjunto de Pareto

Na Figura 6.1, é possível observar os dados de uma instância de teste e o resultado obtido para uma das soluções do conjunto de Pareto. Na parte superior esquerda, o gráfico consolida todas as rotas a serem percorridas durante o horizonte de planejamento. A parte superior direita contém 6 gráficos, um para cada dia do horizonte de planejamento, com a respectiva rota. Para cada gráfico, são exibidas duas informações relativas ao transporte: CD (Capacidade Disponível) e CT (Custo de Transporte).

A parte inferior da Figura 6.1 contém três tabelas. A tabela à esquerda da Figura 6.1 possui três colunas cujas linhas se referem aos clientes (exceto a última) e apresentam os seguintes dados:

- r : consumo do cliente;
- U : nível máximo de estoque no cliente;
- h : custo unitário de estoque no cliente.

A última linha apresenta os seguintes dados (da esquerda para a direita):

- Produção do fornecedor;
- Capacidade de transporte máxima;
- Custo unitário de estoque no fornecedor.

A tabela central na Figura 6.1 possui 7 colunas com dados de estoque nos clientes. A primeira coluna (I_0) exibe o estoque inicial e as demais colunas ($I_1, I_2, I_3, I_4, I_5, I_6$), o estoque para cada dia do horizonte de planejamento, de acordo com a solução obtida. A última linha mostra o somatório das colunas.

A tabela à direita na Figura 6.1 possui 6 colunas ($R_1, R_2, R_3, R_4, R_5, R_6$) com os dados da solução de reabastecimento dos clientes. A última linha mostra o somatório das colunas.

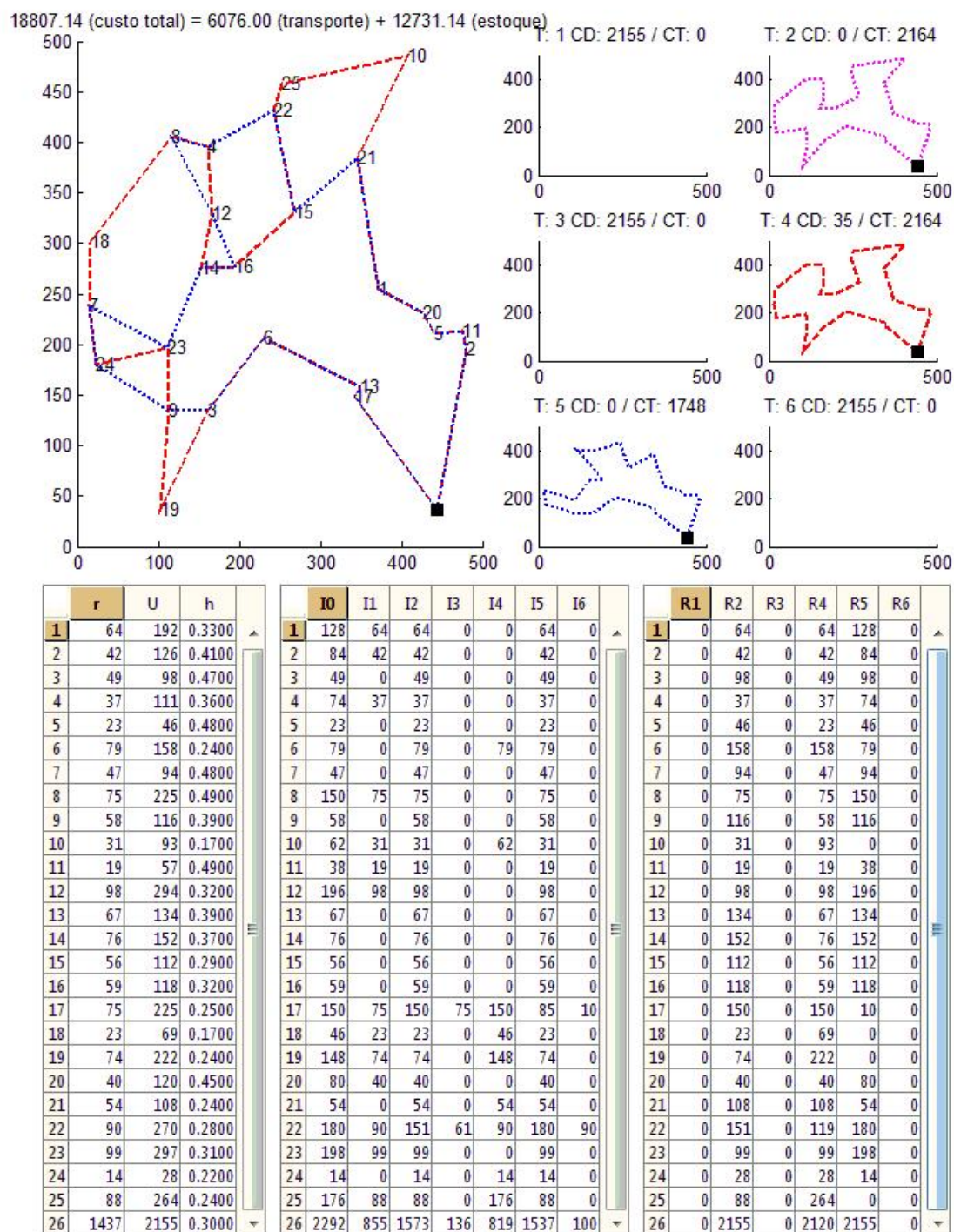


Fig. 6.1: Dados e solução da instância 5, com horizonte de planejamento de 6 dias, custo de estoque alto e 25 clientes

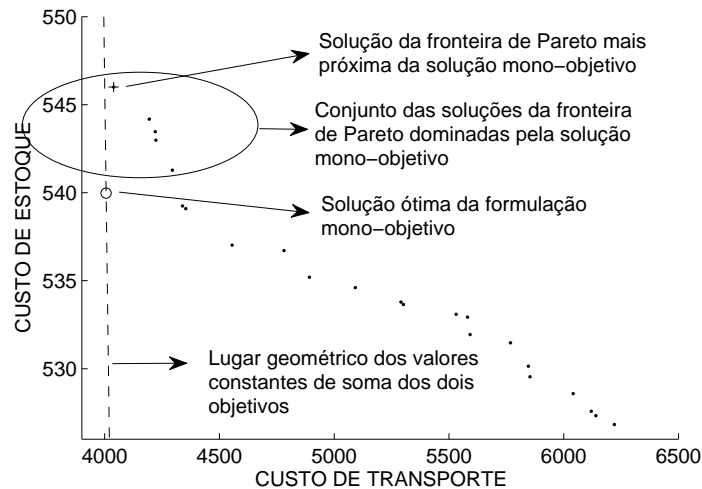


Fig. 6.2: Conjunto de Pareto obtido e solução mono-objetivo para a instância 5, com horizonte de planejamento de 6 dias, custo de estoque baixo e 10 clientes

6.2 Resultados para as instâncias propostas na literatura

A Figura 6.2 mostra as soluções obtidas no espaço dos objetivos e a solução mono-objetivo, referente à instância 5, com horizonte de planejamento de 6 dias, custo de estoque baixo e 10 clientes. As soluções representadas por pontos correspondem ao conjunto de Pareto obtido pela abordagem proposta. A solução indicada com o símbolo \circ corresponde à solução mono-objetivo obtida da literatura. E a solução indicada com $+$ corresponde à solução do conjunto de Pareto mais próxima da solução mono-objetivo no espaço dos objetivos. É possível analisar também o número de soluções do conjunto obtido que são dominadas pela solução mono-objetivo. No exemplo, são 5 as soluções dominadas pela solução mono-objetivo. A fração entre o número de soluções da fronteira de Pareto dominada pela solução mono-objetivo e o número total de soluções da fronteira de Pareto representa uma adaptação da métrica de cobertura, dada pela equação (3.8). Na Figura 6.2, o número de soluções obtidas é 23 e o número de soluções dominadas pela solução ótima mono-objetivo é 5, resultado na fração 0,2174 ou 21,74%.

A Figura 6.2 apresenta um bom espalhamento e o aspecto aparentemente ruidoso da fronteira se deve ao fato do problema ser de otimização discreta. A reta tracejada representa o limitante inferior (do inglês, *lower bound*) para a fronteira de Pareto ótima. A soma dos valores das coordenadas de cada ponto dessa reta é constante e igual ao custo total da solução ótima da formulação mono-objetivo e, conseqüentemente, inclui a solução ótima.

Na Figura 6.3, refere-se à instância 4, com horizonte de planejamento de 6 dias, custo de estoque alto e 15 clientes. Pode-se observar que a solução mono-objetivo domina apenas a solução 'a'. Pode-

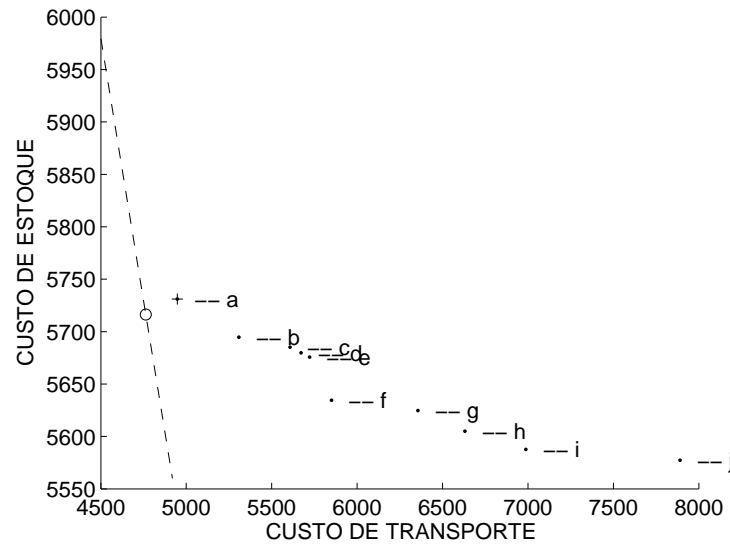


Fig. 6.3: Conjunto de Pareto e solução mono-objetivo para a instância 4, com horizonte de planejamento de 6 dias, alto custo de estoque e 15 clientes

se observar também que as soluções estão bem espalhadas ao longo da fronteira de soluções não-dominadas.

Em relação à diversidade das soluções, foram selecionadas para visualização 6 soluções-candidatas presentes na fronteira de soluções não-dominadas da Figura 6.3, as quais estão apresentadas na Figura 6.4. Nota-se que as soluções diferem em relação a:

- Quantidade de dias em que ocorre algum reabastecimento, ou seja, quantidade de dias em que o veículo é utilizado. Na solução (b), existe reabastecimento em 5 dias. Nas soluções (d) e (g), os dias de reabastecimento são 4, enquanto nas soluções (e), (h) e (j) ocorre reabastecimento em todos os 6 dias.
- Variação dos dias em que ocorre algum reabastecimento. Ambas as soluções (d) e (g) apresentam 4 dias de reabastecimento, sendo que na solução (d) os dias sem reabastecimento são o primeiro e o quinto dia do horizonte de planejamento, enquanto na solução (g), ocorre no primeiro e quarto dia.
- Conjunto de clientes visitados nos dias em que ocorre reabastecimento. Tomando como exemplo o terceiro dia (T3) de cada solução, é fácil identificar pelos diferentes formatos das rotas que há grande diversidade no conjunto de clientes reabastecidos neste dia.

A diversidade de soluções permite que uma solução possa ser trocada por outra com um compromisso de custos equivalente. Suponha, por exemplo, que no dia T_1 o veículo de transporte não possa

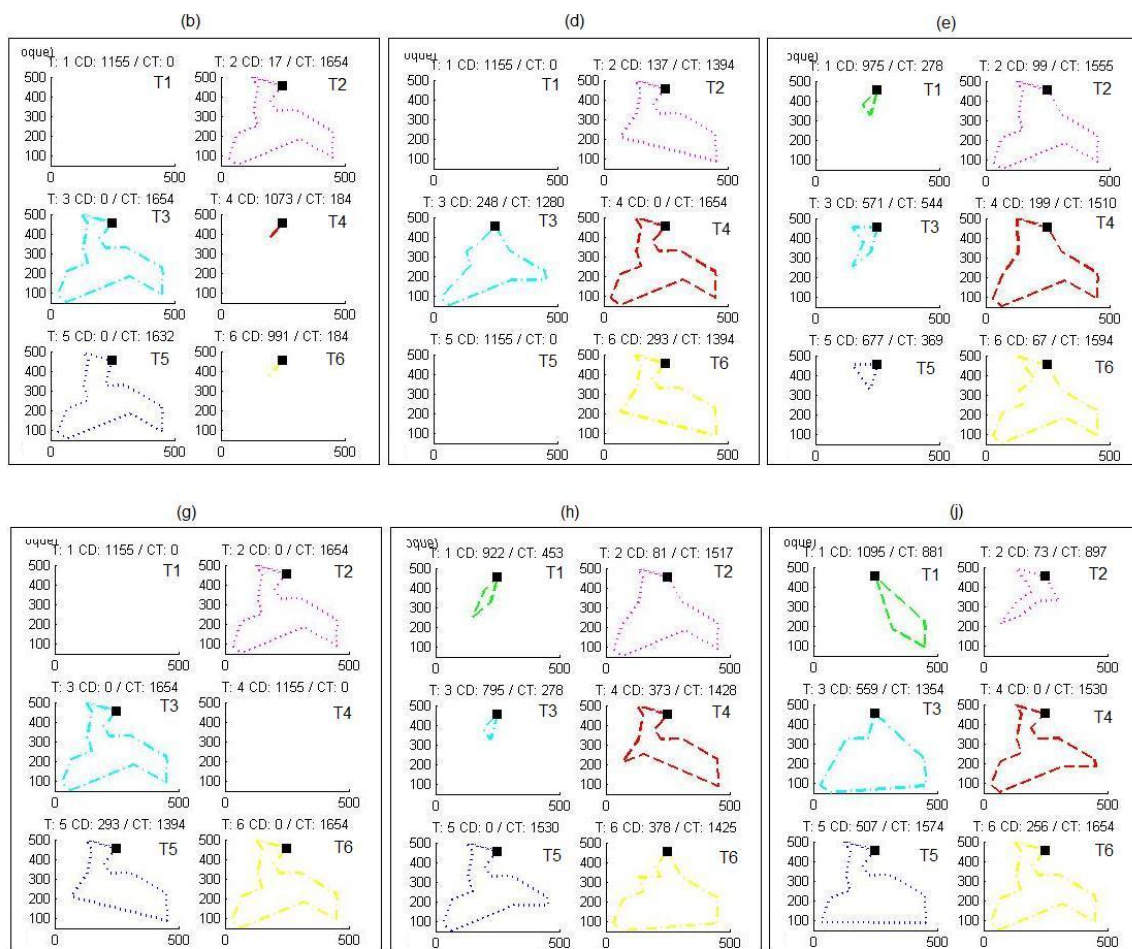


Fig. 6.4: Amostragem de 6 soluções do conjunto de Pareto da Figura 6.3

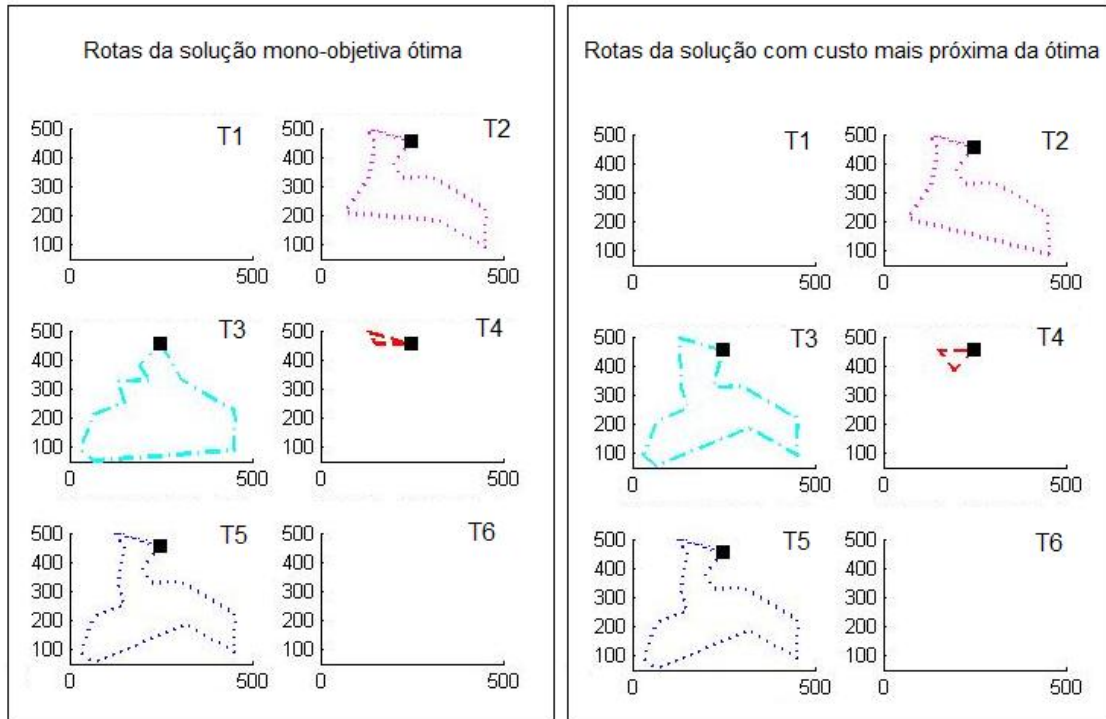


Fig. 6.5: Solução mono-objetivo e solução da fronteira de Pareto mais próxima

ser utilizado. Observando-se os resultados na Figura 6.3, tem-se três soluções ((b), (d) e (g)) em que não há clientes para serem abastecidos nesse dia. Em outro cenário, onde um determinado cliente não pode ser abastecido em um determinado dia, existe grande probabilidade de se encontrar uma solução em que essa restrição seja contemplada.

Outra constatação derivada da análise dos resultados obtidos é que as soluções que apresentam custo total mais baixo, ou seja, as que apresentam melhor desempenho, considerando-se a formulação mono-objetivo, são as que concentram o maior número de reabastecimentos em poucos dias. Outro resultado mais óbvio mostra que, quando a limitação de capacidade de transporte obriga alguns poucos clientes a serem abastecidos em outro dia, as melhores soluções são obtidas quando esses clientes se localizam próximos ao depósito. Um exemplo pode ser observado na Figura 6.4, referente à instância 4, com horizonte de planejamento de 6 dias, custo de estoque alto e 15 clientes.

A Figura 6.5 mostra a comparação entre as rotas da solução ótima da formulação mono-objetivo (solução indicada com o símbolo \circ na Figura 6.3) e as rotas da solução do conjunto de Pareto com a menor distância euclidiana em relação à mono-objetivo (solução indicada com o símbolo $+$ na Figura 6.3).

Um caso em que a solução mono-objetivo coincide com uma das soluções da fronteira de soluções não-dominadas pode ser observado na Figura 6.6, referente à instância 3, com horizonte de

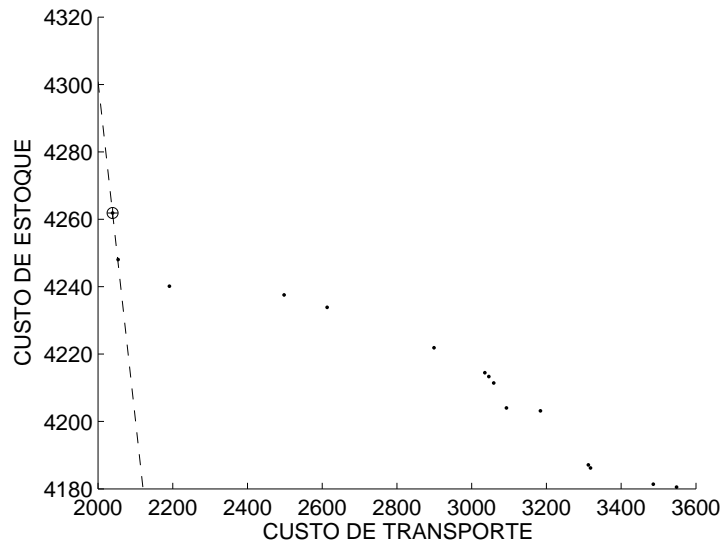


Fig. 6.6: Conjunto de Pareto inclui a solução mono-objetivo, para a instância 3, com horizonte de planejamento de 3 dias, alto custo de estoque e 15 clientes

planejamento de 3 dias, custo de estoque alto e 15 clientes.

As tabelas 6.1, 6.2, 6.3 e 6.4 mostram os resultados obtidos para todas as 160 instâncias de teste. São apresentados os seguintes dados:

- Instância: nome da instância;
- Custo da sol mono-obj:
 - Transporte: custo de transporte;
 - Estoque: custo de estoque.
- Custo da melhor solução: custo da solução do conjunto de Pareto obtido mais próxima da solução mono-objetivo:
 - Transporte: custo de transporte;
 - Estoque: custo de estoque.
- Distância: distância euclidiana no espaço dos objetivos entre a solução mono-objetivo e a solução do conjunto de Pareto mais próxima;
- % distância: porcentagem que corresponde à fração entre a distância do item anterior sobre a distância da solução mono-objetivo até a origem;

- # sol: número de soluções do conjunto de Pareto
- % sol dominadas: porcentagem das soluções do conjunto de Pareto dominadas pela solução mono-objetivo;
- Espalhamento: métrica de diversidade dada pela equação (3.9).

Tab. 6.1: Resultados para instâncias com $H = 3$ e $h_i \in [0, 01; 0, 05]$

Instância	Custo da sol mono-obj		Custo da melhor solução		Distância	% distância	# sol	% sol dominadas	Espalhamento
	Estoque	Transporte	Estoque	Transporte					
abs1n5.dat	94.92	1141.00	94.92	1141.00	-	-	8	0.00	157.37
abs2n5.dat	86.66	902.00	86.66	902.00	-	-	20	0.00	21.16
abs3n5.dat	136.02	1622.00	136.02	1622.00	-	-	9	0.00	37.64
abs4n5.dat	64.29	1333.00	64.29	1333.00	-	-	16	0.00	32.88
abs5n5.dat	131.42	868.00	131.42	868.00	-	-	4	0.00	46.08
abs1n10.dat	303.07	1440.00	303.07	1440.00	-	-	12	0.00	54.42
abs2n10.dat	255.25	1974.00	255.25	1974.00	-	-	9	0.00	116.75
abs3n10.dat	240.14	1631.00	240.14	1631.00	-	-	8	0.00	84.85
abs4n10.dat	243.00	1530.00	243.00	1530.00	-	-	14	0.00	77.63
abs5n10.dat	310.18	1628.00	310.18	1628.00	-	-	13	0.00	51.57
abs1n15.dat	381.04	1750.00	381.04	1750.00	-	-	11	0.00	161.88
abs2n15.dat	368.58	1763.00	368.58	1763.00	-	-	15	0.00	121.81
abs3n15.dat	424.68	2039.00	424.68	2039.00	-	-	15	0.00	244.32
abs4n15.dat	313.94	1838.00	313.94	1838.00	-	-	19	0.00	79.31
abs5n15.dat	303.59	1857.00	303.59	1857.00	-	-	8	0.00	132.81
abs1n20.dat	507.32	1760.00	507.32	1760.00	-	-	16	0.00	84.91
abs2n20.dat	514.90	1983.00	514.90	1983.00	-	-	11	0.00	104.80
abs3n20.dat	533.48	2057.00	533.48	2057.00	-	-	16	0.00	134.37
abs4n20.dat	426.31	2696.00	426.67	2877.00	181.00	6.63	17	5.88	56.51
abs5n20.dat	557.90	2292.00	557.90	2292.00	-	-	11	0.00	69.92
abs1n25.dat	595.92	2245.00	595.92	2245.00	-	-	21	0.00	84.50
abs2n25.dat	637.56	2377.00	637.56	2377.00	-	-	15	6.67	185.11
abs3n25.dat	706.40	2344.00	706.40	2344.00	-	-	11	0.00	163.76
abs4n25.dat	599.67	2479.00	600.39	2500.00	21.01	0.82	18	11.11	117.79
abs5n25.dat	790.96	2164.00	790.96	2164.00	-	-	19	0.00	75.69
abs1n30.dat	956.78	2471.00	956.78	2471.00	-	-	15	0.00	129.29
abs2n30.dat	844.94	2484.00	844.94	2484.00	-	-	21	0.00	49.14
abs3n30.dat	965.86	2506.00	965.86	2506.00	-	-	20	0.00	54.61
abs4n30.dat	711.48	2610.00	711.81	2647.00	37.00	1.37	26	3.85	55.72
abs5n30.dat	763.60	2151.00	763.60	2151.00	-	-	14	0.00	130.00
abs1n35.dat	914.12	2432.00	914.12	2432.00	-	-	15	0.00	45.73
abs2n35.dat	770.71	2771.00	771.53	2977.00	206.00	7.16	17	5.88	125.86
abs3n35.dat	1101.78	2710.00	1101.78	2710.00	-	-	13	0.00	53.99
abs4n35.dat	791.34	2438.00	791.34	2438.00	-	-	27	0.00	154.43
abs5n35.dat	835.26	2480.00	835.26	2480.00	-	-	17	0.00	29.58
abs1n40.dat	1064.14	2638.00	1064.14	2638.00	-	-	21	0.00	79.42
abs2n40.dat	833.09	2999.00	833.09	2999.00	-	-	15	0.00	46.77
abs3n40.dat	1083.62	2791.00	1083.62	2791.00	-	-	21	0.00	58.37
abs4n40.dat	876.80	2658.00	876.80	2658.00	-	-	19	0.00	58.85
abs5n40.dat	1050.46	2525.00	1050.46	2525.00	-	-	20	0.00	28.84
abs1n45.dat	1130.86	2820.00	1130.86	2820.00	-	-	27	0.00	37.80
abs2n45.dat	1051.72	2651.00	1051.72	2651.00	-	-	17	0.00	154.10
abs3n45.dat	1205.04	2763.00	1205.04	2763.00	-	-	24	0.00	104.12
abs4n45.dat	1073.26	2925.00	1073.26	2925.00	-	-	23	0.00	49.91
abs5n45.dat	1092.54	2625.00	1094.12	2640.00	15.08	0.53	23	17.39	80.86
abs1n50.dat	1166.18	2881.00	1166.18	2881.00	-	-	21	0.00	56.36
abs2n50.dat	1171.96	3341.00	1171.96	3341.00	-	-	17	0.00	86.95
abs3n50.dat	1202.44	3249.00	1202.44	3350.00	101.00	2.92	16	6.25	63.12
abs4n50.dat	1354.84	3051.00	1354.84	3051.00	-	-	21	0.00	42.40
abs5n50.dat	1273.37	2945.00	1271.70	2965.00	20.07	0.63	20	0.00	137.17

Tab. 6.2: Resultados para instâncias com $H = 3$ e $h_i \in [0, 1; 0, 5]$

Instância	Custo da sol mono-obj		Custo da melhor solução		Distância	% distância	# sol	% sol dominadas	Espalhamento
	Estoque	Transporte	Estoque	Transporte					
abs1n5.dat	967.34	1141.00	967.34	1141.00	-	-	10	0.00	117.89
abs2n5.dat	865.06	902.00	865.06	902.00	-	-	18	0.00	24.45
abs3n5.dat	1351.00	1622.00	1351.00	1622.00	-	-	9	0.00	38.45
abs4n5.dat	648.04	1333.00	648.04	1333.00	-	-	19	0.00	31.50
abs5n5.dat	1302.04	868.00	1302.04	868.00	-	-	5	0.00	54.96
abs1n10.dat	3070.61	1440.00	3070.61	1440.00	-	-	15	0.00	51.83
abs2n10.dat	2530.61	1974.00	2530.61	1974.00	-	-	9	0.00	165.54
abs3n10.dat	2400.40	1631.00	2400.40	1631.00	-	-	14	0.00	88.48
abs4n10.dat	2403.46	1530.00	2403.46	1530.00	-	-	15	0.00	28.97
abs5n10.dat	3081.79	1628.00	3081.79	1628.00	-	-	17	0.00	26.38
abs1n15.dat	3839.70	1750.00	3839.70	1750.00	-	-	13	0.00	84.85
abs2n15.dat	3680.34	1763.00	3680.34	1763.00	-	-	17	0.00	124.66
abs3n15.dat	4261.86	2039.00	4261.86	2039.00	-	-	15	0.00	48.40
abs4n15.dat	3139.58	1838.00	3139.58	1838.00	-	-	15	0.00	87.24
abs5n15.dat	3010.53	1857.00	3010.53	1857.00	-	-	8	12.50	96.27
abs1n20.dat	5099.02	1760.00	5099.02	1760.00	-	-	12	0.00	50.93
abs2n20.dat	5104.74	1983.00	5104.74	1983.00	-	-	13	0.00	102.66
abs3n20.dat	5297.68	2057.00	5297.68	2057.00	-	-	16	0.00	186.97
abs4n20.dat	4256.79	2696.00	4269.37	2787.00	91.87	1.82	14	7.14	235.04
abs5n20.dat	5582.26	2292.00	5582.26	2292.00	-	-	17	0.00	107.96
abs1n25.dat	5982.86	2245.00	5982.86	2245.00	-	-	21	0.00	274.91
abs2n25.dat	6388.72	2377.00	6388.72	2377.00	-	-	16	12.50	103.74
abs3n25.dat	7038.42	2344.00	7038.42	2344.00	-	-	14	0.00	94.86
abs4n25.dat	5973.93	2479.00	5991.04	2525.00	49.08	0.76	19	10.53	112.48
abs5n25.dat	7917.42	2164.00	7917.42	2164.00	-	-	20	0.00	108.96
abs1n30.dat	9595.86	2471.00	9595.86	2471.00	-	-	15	0.00	188.43
abs2n30.dat	8457.32	2484.00	8457.32	2484.00	-	-	21	0.00	119.21
abs3n30.dat	9616.36	2506.00	9616.36	2506.00	-	-	18	0.00	112.59
abs4n30.dat	7077.10	2610.00	7076.88	2692.00	82.00	1.09	25	0.00	146.05
abs5n30.dat	7622.90	2151.00	7622.90	2151.00	-	-	16	0.00	56.02
abs1n35.dat	9227.88	2432.00	9227.88	2432.00	-	-	12	0.00	251.19
abs2n35.dat	7695.80	2771.00	7693.09	2987.00	216.02	2.64	17	0.00	93.49
abs3n35.dat	11066.46	2710.00	11066.46	2710.00	-	-	25	0.00	47.95
abs4n35.dat	7869.40	2438.00	7869.40	2438.00	-	-	21	0.00	86.78
abs5n35.dat	8367.82	2480.00	8367.82	2480.00	-	-	19	0.00	89.66
abs1n40.dat	10726.92	2638.00	10726.92	2638.00	-	-	16	0.00	140.32
abs2n40.dat	8318.85	2999.00	8318.85	2999.00	-	-	22	0.00	137.29
abs3n40.dat	10807.94	2791.00	10807.94	2791.00	-	-	26	0.00	80.72
abs4n40.dat	8637.39	2716.00	8730.62	2658.00	109.80	1.21	20	55.00	143.45
abs5n40.dat	10545.18	2525.00	10545.18	2525.00	-	-	21	0.00	70.60
abs1n45.dat	11359.10	2820.00	11359.10	2820.00	-	-	27	0.00	111.05
abs2n45.dat	10491.22	2651.00	10491.22	2651.00	-	-	23	0.00	229.77
abs3n45.dat	12080.60	2763.00	12080.60	2763.00	-	-	20	0.00	75.65
abs4n45.dat	10649.50	2925.00	10649.50	2925.00	-	-	19	0.00	56.91
abs5n45.dat	10962.26	2625.00	10962.26	2625.00	-	-	23	0.00	80.11
abs1n50.dat	11694.30	2883.00	11694.30	2883.00	-	-	23	0.00	231.90
abs2n50.dat	11620.64	3381.00	11692.80	3372.00	72.72	0.60	26	26.92	166.63
abs3n50.dat	12030.49	3249.00	12030.49	3249.00	-	-	23	4.35	211.17
abs4n50.dat	13466.00	3051.00	13466.00	3051.00	-	-	29	0.00	103.46
abs5n50.dat	12733.67	2945.00	12720.16	2965.00	24.14	0.18	14	0.00	228.12

Tab. 6.3: Resultados para instâncias com $H = 6$ e $h_i \in [0, 01; 0, 05]$

Instância	Custo da sol mono-obj		Custo da melhor solução		Distância	% distância	# sol	% sol dominadas	Espalhamento
	Estoque	Transporte	Estoque	Transporte					
abs1n5.dat	284.30	2903.00	286.65	2907.00	4.64	0.16	18	16.67	73.05
abs2n5.dat	261.92	2304.00	259.60	2332.00	28.10	1.21	20	5.00	87.52
abs3n5.dat	238.83	4251.00	241.58	4251.00	2.75	0.06	15	13.33	82.23
abs4n5.dat	210.35	2964.00	210.83	2964.00	0.48	0.02	21	9.52	55.40
abs5n5.dat	236.10	2031.00	236.10	2031.00	-	-	12	0.00	55.25
abs1n10.dat	476.53	3665.00	476.85	3726.00	61.00	1.65	13	0.00	105.33
abs2n10.dat	367.63	4677.00	370.37	4705.00	28.13	0.60	23	21.74	79.02
abs3n10.dat	425.83	4081.00	427.61	4276.00	195.01	4.75	19	10.53	57.62
abs4n10.dat	412.53	4411.00	414.06	4411.00	1.53	0.03	14	7.14	105.66
abs5n10.dat	539.98	4006.00	546.00	4039.00	33.54	0.83	23	21.74	60.83
abs1n15.dat	735.08	4654.00	737.50	4805.00	151.02	3.21	9	22.22	88.55
abs2n15.dat	715.47	4703.00	718.32	4742.00	39.10	0.82	19	10.53	199.76
abs3n15.dat	819.68	5078.00	820.34	5324.00	246.00	4.78	19	10.53	99.20
abs4n15.dat	572.01	4763.00	571.31	4965.00	202.00	4.21	11	0.00	288.69
abs5n15.dat	560.51	4492.00	564.60	4679.00	187.04	4.13	24	20.83	76.08
abs1n20.dat	906.04	5208.00	907.14	5419.00	211.00	3.99	13	7.69	110.30
abs2n20.dat	956.31	5001.00	959.74	5262.00	261.02	5.13	8	37.50	465.79
abs3n20.dat	843.06	5941.00	841.18	6075.00	134.01	2.23	12	0.00	248.75
abs4n20.dat	788.54	6521.00	788.53	6783.00	262.00	3.99	15	13.33	239.25
abs5n20.dat	955.82	6006.00	956.72	6414.00	408.00	6.71	13	7.69	121.78
abs1n25.dat	935.06	6117.00	940.12	6449.00	332.04	5.37	20	15.00	185.05
abs2n25.dat	1025.75	6206.00	1024.77	6662.00	456.00	7.25	16	0.00	84.67
abs3n25.dat	1155.57	6359.00	1158.99	6867.00	508.01	7.86	14	7.14	149.33
abs4n25.dat	976.08	6486.00	979.42	6818.00	332.02	5.06	18	61.11	93.41
abs5n25.dat	1275.40	5773.00	1272.07	6057.00	284.02	4.80	9	0.00	396.38
abs1n30.dat	1640.73	6412.00	1628.62	7189.00	777.09	11.74	8	0.00	326.32
abs2n30.dat	1360.99	6269.00	1364.37	6850.00	581.01	9.06	19	10.53	77.16
abs3n30.dat	1679.21	6457.00	1687.78	6907.00	450.08	6.75	16	6.25	170.63
abs4n30.dat	1119.49	6383.00	1121.59	7268.00	885.00	13.66	17	11.76	107.72
abs5n30.dat	1281.65	5989.00	1276.75	6397.00	408.03	6.66	8	0.00	153.22

Tab. 6.4: Resultados para instâncias com $H = 6$ e $h_i \in [0, 1; 0, 5]$

<i>Instância</i>	<i>Custo da sol mono-obj</i>		<i>Custo da melhor solução</i>		<i>Distância</i>	<i>% distância</i>	<i># sol</i>	<i>% sol dominadas</i>	<i>Espalhamento</i>
	<i>Estoque</i>	<i>Transporte</i>	<i>Estoque</i>	<i>Transporte</i>					
abs1n10.dat	4815.17	3665.00	4817.89	3731.00	66.06	1.09	23	0.00	72.22
abs2n10.dat	3670.44	4677.00	3688.65	4705.00	33.40	0.56	23	17.39	178.47
abs3n10.dat	4240.68	4081.00	4258.06	4222.00	142.07	2.41	27	14.81	43.16
abs4n10.dat	4063.26	4411.00	4086.50	4411.00	23.24	0.39	15	13.33	113.35
abs5n10.dat	5380.03	4006.00	5435.35	4039.00	64.41	0.96	17	23.53	129.93
abs1n15.dat	7343.56	4709.00	7413.39	4833.00	142.31	1.63	10	20.00	95.97
abs2n15.dat	7089.55	4734.00	7129.04	5003.00	271.88	3.19	15	46.67	125.38
abs3n15.dat	8227.71	5078.00	8240.87	5324.00	246.35	2.55	25	12.00	50.53
abs4n15.dat	5716.29	4763.00	5730.99	4948.00	185.58	2.49	10	10.00	239.36
abs5n15.dat	5561.09	4493.00	5616.56	4643.00	159.93	2.24	26	15.38	85.98
abs1n20.dat	9024.53	5242.00	9110.16	5419.00	196.63	1.88	13	30.77	76.42
abs2n20.dat	9461.84	5016.00	9530.31	5325.00	316.50	2.96	15	26.67	43.75
abs3n20.dat	8378.35	5941.00	8339.22	6118.00	181.27	1.76	5	0.00	328.41
abs4n20.dat	7835.27	6555.00	7898.14	6926.00	376.29	3.68	15	60.00	69.51
abs5n20.dat	9550.70	6006.00	9495.63	6574.00	570.66	5.06	8	12.50	105.20
abs1n25.dat	9341.66	6146.00	9445.48	6438.00	309.91	2.77	12	66.67	139.37
abs2n25.dat	10295.46	6207.00	10267.42	6662.00	455.86	3.79	9	0.00	227.30
abs3n25.dat	11454.35	6379.00	11465.62	6884.00	505.13	3.85	12	16.67	176.76
abs4n25.dat	9707.96	6486.00	9803.16	6953.00	476.60	4.08	18	55.56	53.47
abs5n25.dat	12768.42	5784.00	12731.14	6076.00	294.37	2.10	10	0.00	210.51
abs1n30.dat	16396.94	6441.00	16341.91	7369.00	929.63	5.28	8	0.00	228.42
abs2n30.dat	13598.76	6278.00	13603.96	7153.00	875.02	5.84	10	10.00	110.35
abs3n30.dat	16579.93	6517.00	16619.65	7133.00	617.28	3.46	12	41.67	297.63
abs4n30.dat	11116.82	6393.00	11147.03	6988.00	595.77	4.65	12	33.33	293.35
abs5n30.dat	12765.81	5966.00	12748.12	6397.00	431.36	3.06	8	0.00	205.03

6.3 Amostragem de soluções

A Figura 6.7 exibe as fronteiras de Pareto para 9 das piores soluções obtidas de acordo com a métrica de convergência, ou seja, que apresentam as maiores distâncias entre a solução ótima mono-objetivo e a solução mais próxima em termos percentuais (ver coluna % distância nas Tabelas 6.1, 6.2, 6.3 e 6.4). A solução ótima da formulação mono-objetivo pode ser identificada pelo símbolo ○. A partir da análise das melhores soluções indicadas a seguir, nota-se que sempre que o conjunto de Pareto contém a solução ótima da formulação mono-objetivo, ela se localiza no extremo superior da fronteira de Pareto. Isso indica que há espaço para melhorias no algoritmo em relação ao custo de transporte. Por inspeção visual, nota-se um bom espalhamento ao longo da fronteira de Pareto. Testes com algumas instâncias do problema mostraram que, à medida que se aumenta o número de iterações do SPEA2, ocorre um melhor preenchimento da fronteira de Pareto, mesmo quando a convergência da solução se estabiliza.

A Figura 6.8 exibe as fronteiras de Pareto para 9 das soluções obtidas em que o conjunto de Pareto contém a solução ótima da formulação mono-objetivo, o que indica uma boa convergência do algoritmo.

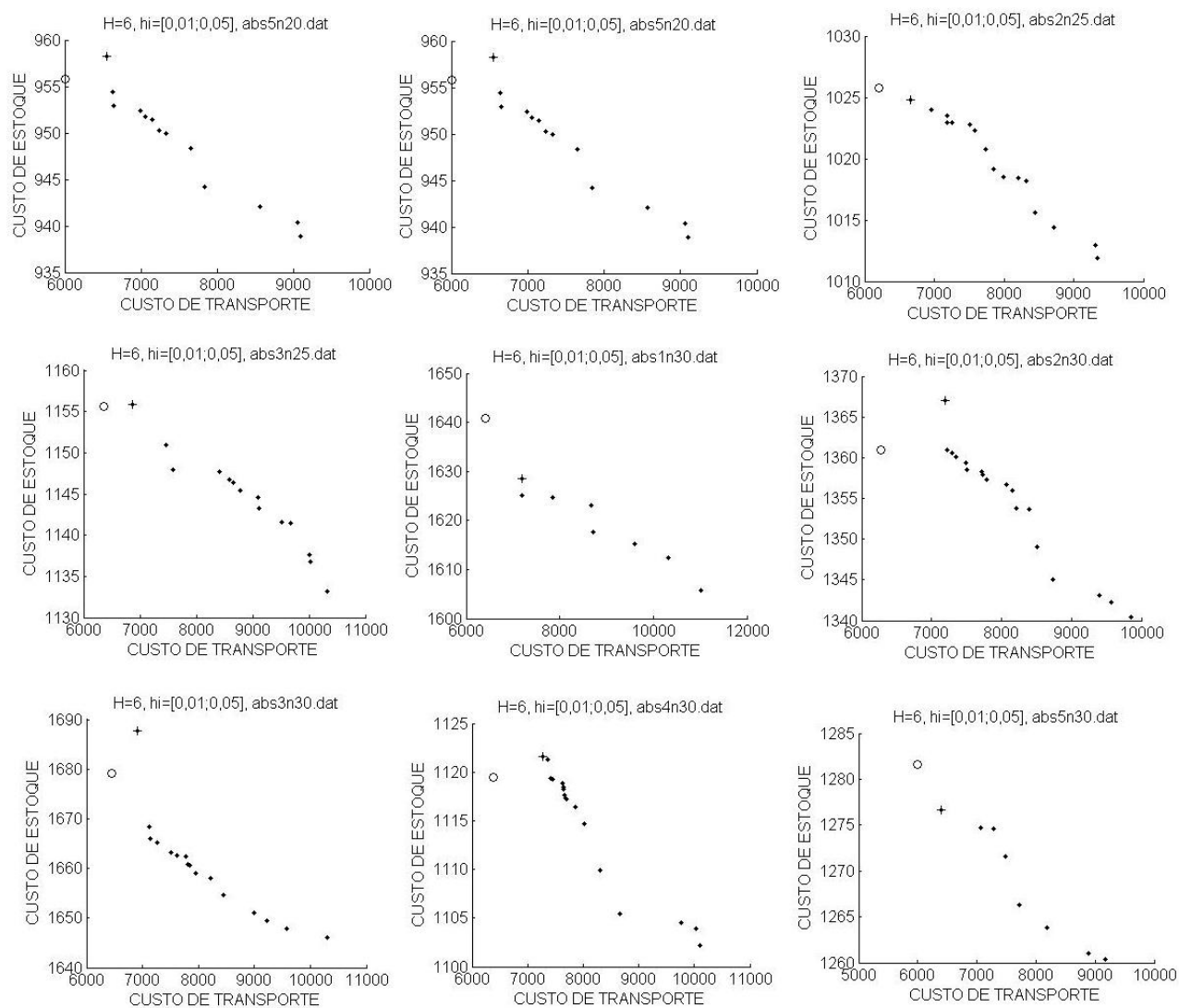


Fig. 6.7: Piores resultados

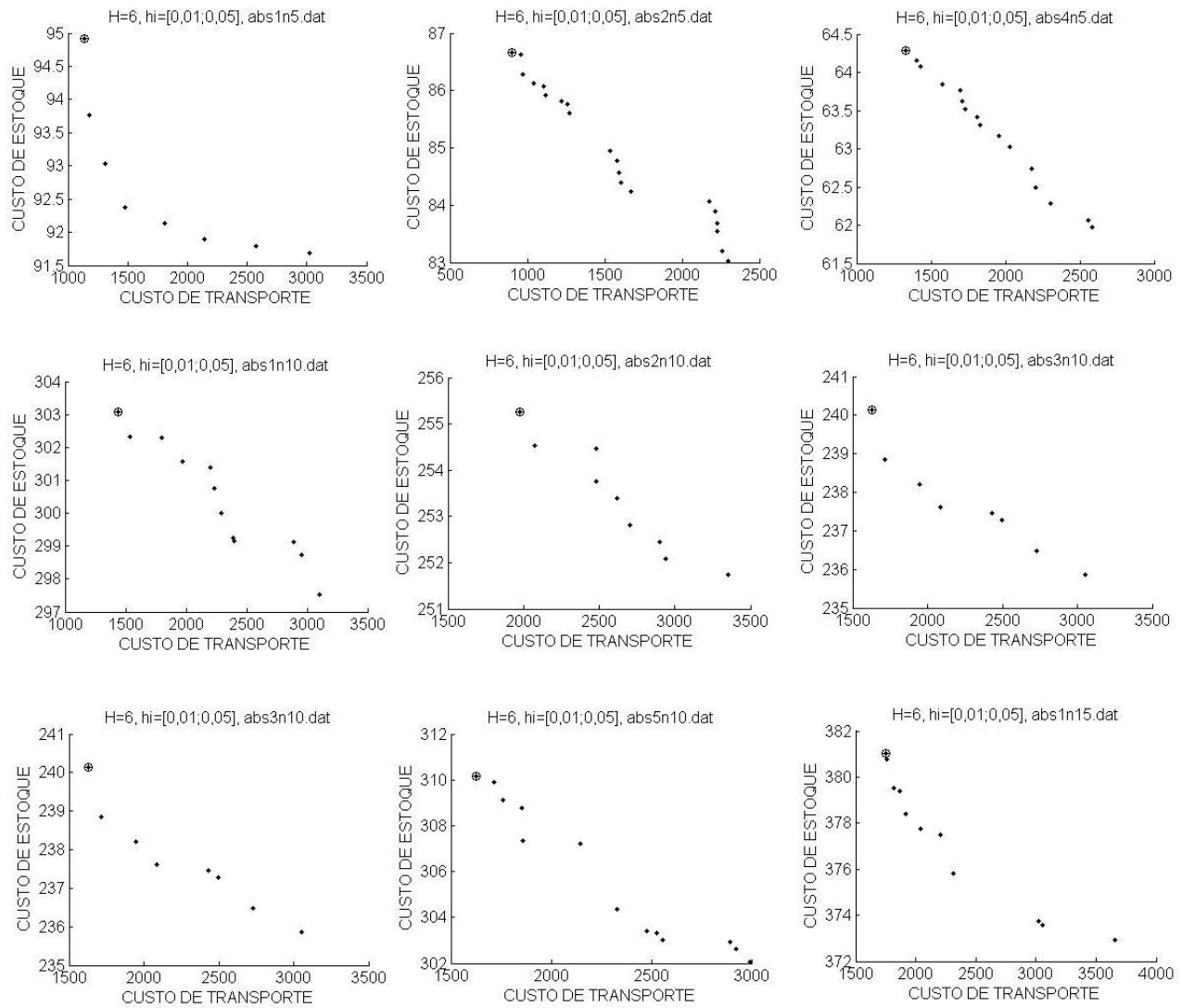


Fig. 6.8: Melhores resultados

6.4 Resultados para as instâncias de grande porte

Foram realizados testes com instâncias geradas unicamente para avaliar a escalabilidade da solução. As instâncias estão descritas na seção 4.6. Nesses casos, não é possível avaliar a convergência do algoritmo porque são instâncias novas propostas para esse trabalho.

A Figura 6.9 mostra o conjunto de Pareto da solução obtida para uma instância com 200 clientes e horizonte de planejamento de 3 dias. Nota-se que a fronteira apresenta um bom espalhamento, mas o algoritmo não conseguiu povoar uma região. Resultados similares foram obtidos em outras execuções do algoritmo para a mesma instância, não sendo possível afirmar se é uma característica da instância ou deficiência do algoritmo em tratar a instância adequadamente.

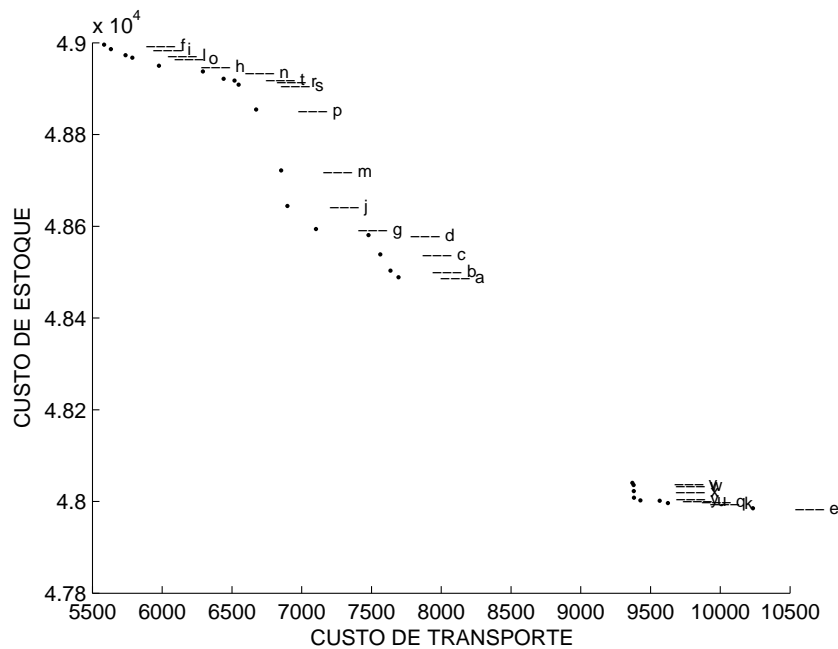


Fig. 6.9: Conjunto de Pareto para uma instância com 200 clientes e horizonte de 3 dias

A Figura 6.10 mostra 4 soluções do conjunto de Pareto da Figura 6.9. A solução com menor custo de transporte está identificada como (f) e a solução com menor custo de estoque está identificada como (e). Nota-se, a partir das rotas, uma grande variedade nas soluções.

A Figura 6.11 mostra o conjunto de Pareto da solução obtida para uma instância com 200 clientes e horizonte de planejamento de 6 dias.

A Figura 6.12 mostra 4 soluções do conjunto de Pareto da Figura 6.11. A solução com menor custo de transporte está identificada como (d) e a solução com menor custo de estoque está identificada como (a). Nota-se, também nesse caso, uma grande variedade nas soluções.

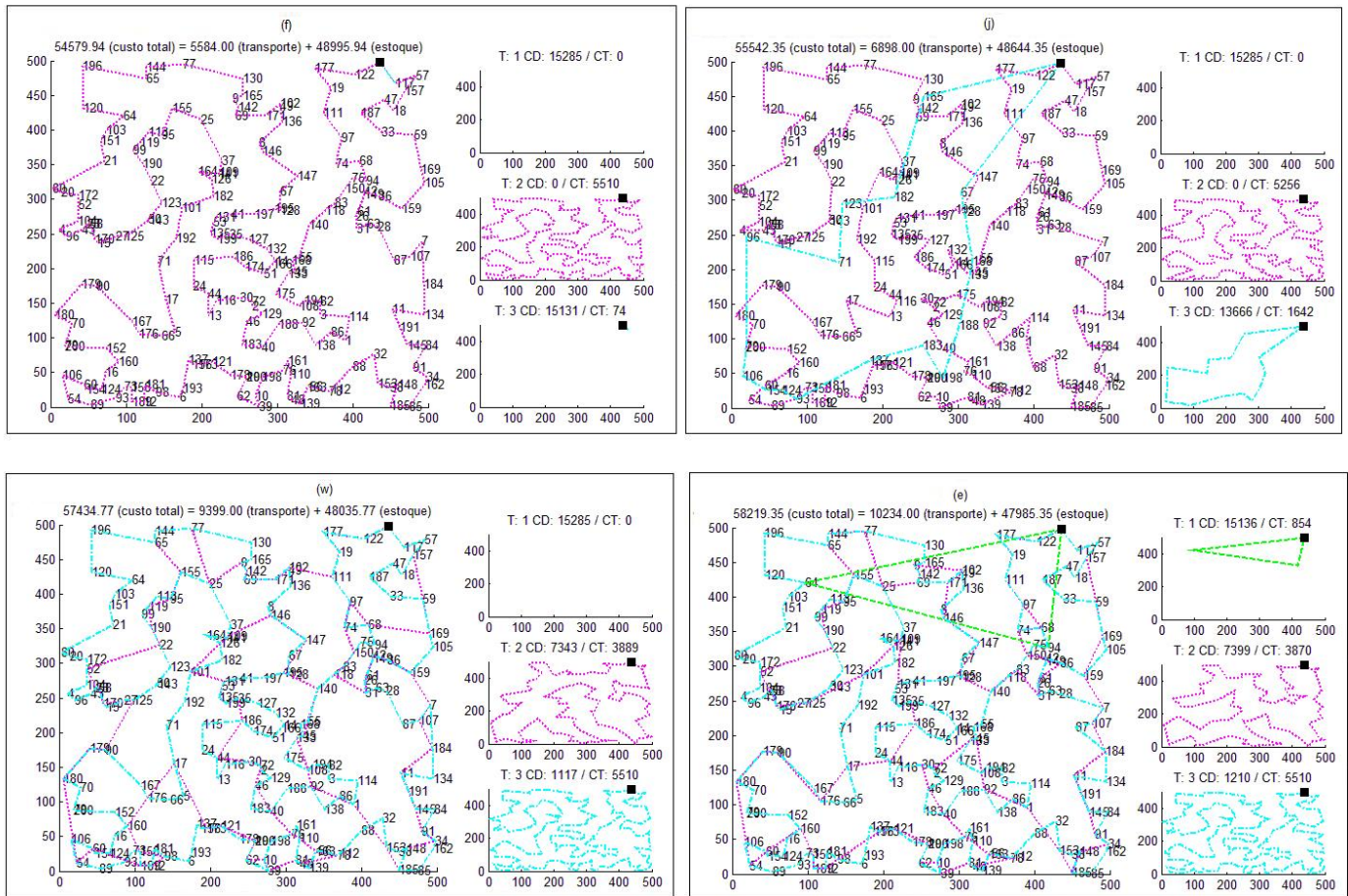


Fig. 6.10: Amostragem de 4 soluções do conjunto de Pareto da Figura 6.9

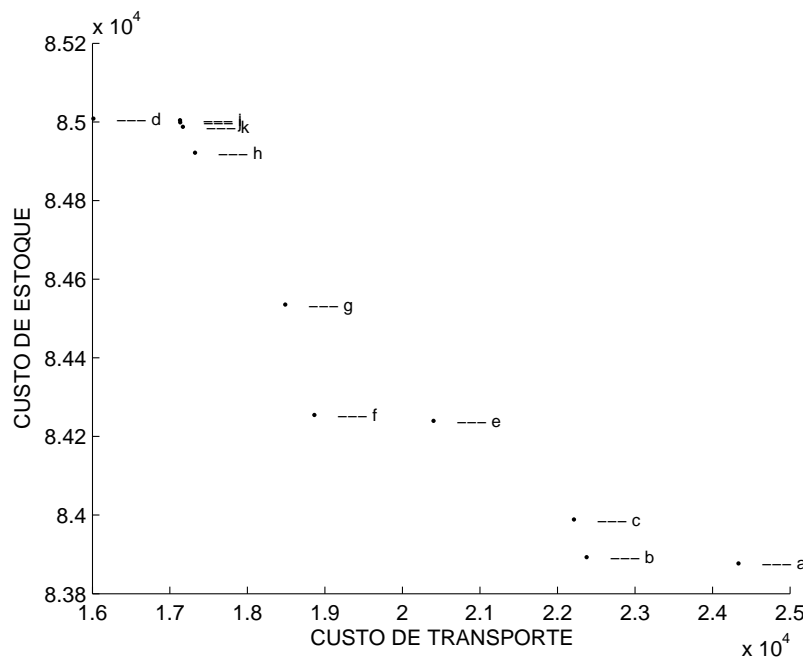


Fig. 6.11: Conjunto de Pareto para instância com 200 clientes e horizonte de 6 dias

6.5 Síntese do capítulo

Neste capítulo, foram apresentados os resultados obtidos a partir da aplicação da abordagem multiobjetivo a um conjunto de instâncias de pequeno e médio porte conhecidas na literatura, para os quais existe solução ótima para a formulação mono-objetivo. Os resultados compreendem métricas de convergência e diversidade. Foram exibidos também os resultados obtidos a partir da aplicação da abordagem proposta a duas instâncias de grande porte. Como não foi possível encontrar na literatura instâncias de grande porte, elas foram geradas aleatoriamente, não sendo possível avaliar a convergência das soluções devido à inexistência de algoritmos exatos capazes de lidar com problemas de grande porte.

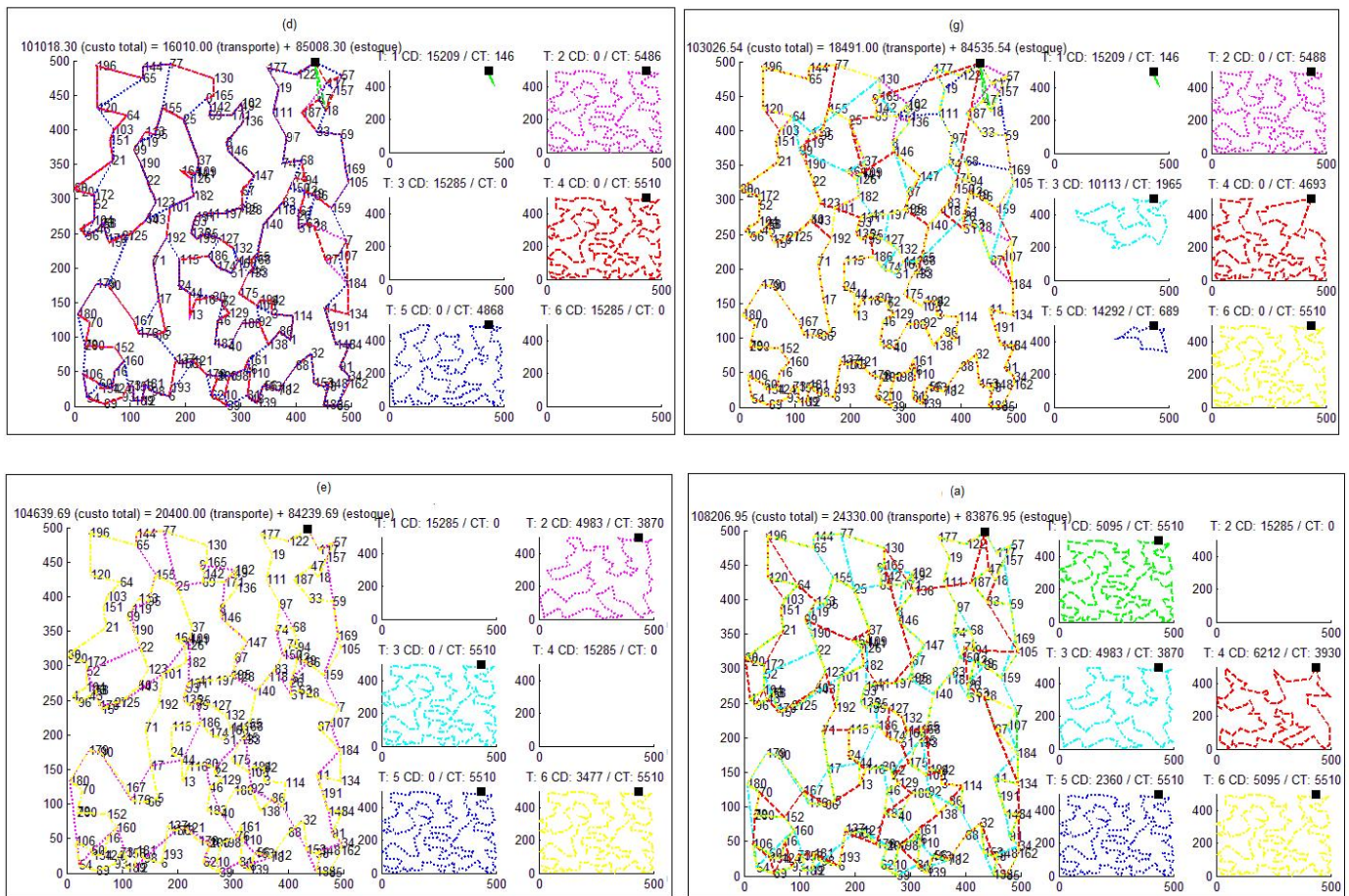


Fig. 6.12: Amostragem de 4 soluções do conjunto de Pareto da Figura 6.11

Capítulo 7

Conclusão e perspectivas

Neste capítulo, são feitas algumas considerações, são apresentadas as principais contribuições desta dissertação e são apontadas algumas linhas de pesquisa relacionadas a possíveis extensões deste trabalho.

7.1 Considerações finais

A principal motivação do trabalho foi a existência de um grande número de problemas reais com suas diferentes nuances e que envolvem a otimização de estoque e roteamento. Nesses casos, os métodos exatos se mostram intratáveis, principalmente pelo número elevado de variáveis quando se tem um grande número de clientes e pela quantidade de restrições que, em geral, são conflitantes e inserem regiões de infactibilidade no espaço de busca. Esse fato tem motivado a pesquisa na área de métodos heurísticos, que possuem a vantagem de serem mais escaláveis e tratarem de forma simples a questão da factibilidade de soluções.

Grande parte dos problemas reais, incluindo o caso particular considerado no trabalho, são inerentemente multiobjetivo. Em muitas abordagens, o problema multiobjetivo é transformado em mono-objetivo, usando-se uma técnica de tomada de decisão que agrega os objetivos em um valor escalar que representa as preferências do decisor. Essa técnica apresenta a desvantagem de exigir que o decisor articule suas preferências *a priori* e a dificuldade de ajustar os pesos das preferências. Ao se converter os objetivos em um valor escalar, muita informação relevante é perdida. A vantagem de uma abordagem multiobjetivo é que o decisor pode articular suas preferências tendo o conjunto de soluções-alternativas em mãos. Dessa forma, uma mudança de interesse do decisor não exige uma nova execução do algoritmo de busca, bastando considerar, dentre as soluções alternativas já obtidas, aquelas que melhor atendem aos novos interesses. Além disso, a abordagem multiobjetivo permite uma maior facilidade na inclusão de critérios que não são naturalmente quantificáveis.

Na área de otimização multiobjetivo, as abordagens populacionais têm se mostrado eficazes, pois provêm uma exploração eficiente do espaço de busca e permitem obter várias soluções não-dominadas, que aproximam a fronteira de Pareto, simultaneamente. A possibilidade de se ter soluções variadas com um custo computacional aceitável é muito importante do ponto de vista prático.

Como principais contribuições do trabalho, pode-se destacar:

- A inclusão de violação de restrições no critério de dominância, que permite uma melhor exploração do espaço de busca, principalmente em instâncias de teste em que as restrições são mais fortes e é difícil encontrar soluções factíveis. A existência de soluções não-factíveis na população e a possibilidade de compará-las permite que os objetivos sejam melhor explorados. O que ocorre na prática é que o número de soluções não-factíveis decresce continuamente ao longo das iterações, restando somente as factíveis.
- O tipo de representação adotada, que contém duas informações: dias de reabastecimento para cada cliente e quantidades de reabastecimento. Esse mapeamento, ao tratar diretamente as quantidades de reabastecimento, permite uma evolução mais rápida do algoritmo.
- Os operadores de mutação e recombinação concebidos, apesar de terem sido implementados visando resolver o problema proposto, podem ser considerados de propósito geral para problemas de estoque e roteamento.
- A inclusão de buscas locais que objetivam substituir as soluções existentes por soluções que as dominem também foi de grande importância para a melhoria do desempenho da abordagem adotada.

A eficiência da heurística proposta pode ser observada pelos resultados obtidos. Em 62,50 % das instâncias de teste, a solução ótima da formulação mono-objetivo do problema não domina nenhuma das soluções do conjunto de Pareto obtido pela abordagem proposta. Em 55,00% das instâncias, uma das soluções do conjunto de Pareto obtido coincide com a solução ótima mono-objetivo. A diferença percentual média entre a solução ótima mono-objetivo e a solução do conjunto de Pareto mais próxima, considerando a distância euclidiana, é de 1,43%.

A análise da diversidade das soluções obtidas permitiu observar que as soluções com custo de transporte mais baixo são as que concentram os reabastecimentos em um número menor de dias. Nos casos em que a limitação de capacidade requer que alguns poucos clientes sejam reabastecidos em outro dia, a escolha de clientes mais próximos do depósito favorece uma diminuição dos custos de transporte. Por outro lado, nas soluções com custo de estoque mais baixo, ocorre uma melhor

distribuição de reabastecimentos ao longo do horizonte de planejamento. Esse é também um dos propósitos da otimização multiobjetivo: levantar soluções alternativas que privilegiam aspectos distintos do problema.

7.2 Perspectivas e pesquisas futuras

O problema que esse trabalho se propõe a resolver trata de uma demanda constante nos clientes. No entanto, em problemas práticos, como por exemplo, o caso de distribuição de numerário em redes ATM ou de distribuição de medicamentos e perfumaria, a demanda é estocástica. Dessa forma, uma extensão natural da abordagem apresentada seria a inclusão de um módulo de predição de consumo para tratar demandas estocásticas. Em problemas com demandas estocásticas, a utilização de horizonte deslizante de planejamento pode apresentar bons resultados, principalmente quando a realimentação dos dados é rápida e precisa. A utilização de horizonte deslizante implica em fazer o planejamento para um período longo de tempo, utilizar apenas a parte inicial do planejamento, re-alimentar a base de dados com a demanda real, replanejar outro horizonte de tempo com base nas informações atualizadas, e assim por diante.

Outro aspecto relevante em problemas reais é a utilização de cenários *WHAT-IF*. Poder-se-ia investigar mecanismos para inclusão *ad hoc* de restrições. A estratégia proposta comporta uma inclusão direta de restrições e o uso de uma abordagem populacional permite que, em poucas iterações (ou mesmo nenhuma), obtenham-se soluções que satisfaçam as novas restrições.

Outra linha de pesquisa seria o uso de algoritmos de estimação de distribuição (EDAs, do inglês *Estimation of Distribution Algorithms*). Os EDAs são um exemplo de meta-heurística populacional que emprega a experiência passada da busca para guiar a geração de novas soluções-candidatas. Ao invés de operadores de mutação e recombinação, modelos probabilísticos são empregados na amostragem do espaço de busca.

No trabalho, optou-se por usar uma meta-heurística mais consolidada, mais especificamente o algoritmo evolutivo SPEA2. No entanto, há espaço também para a investigação de outras meta-heurísticas bioinspiradas capazes de lidar com os aspectos multimodais e multiobjetivos do problema em estudo.

Referências Bibliográficas

- Archetti, C., Bertazzi, L., Laporte, G., & Speranza, M. G. 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation science*, **41**(August), 382–391.
- Archetti, C., Bertazzi, L., Hertz, A., & Speranza, M. G. 2011. A hybrid heuristic for an inventory-routing problem. *Inform's journal on computing*.
- Baita, F., Ukovich, W., Pesenti, R., & Favaretto, D. 1998. Dynamic routing-and-inventory problems: a review. *Transportation research part a: Policy and practice*, **32**(8), 585–598.
- Ball, M. O. 1998. Allocation/Routing: models and algorithms. *Vehicle routing: Methods and studies*, 199–221.
- Bard, J., Huang, L., Jaillet, P., & Dror, M. 1998. A decomposition approach to the inventory routing problem with satellite facilities. *Transportation science*, 189–203.
- Bertazzi, L., Paletta, G., & Speranza, M. G. 2005. Minimizing the total cost in an integrated vendor–managed inventory system. *Journal of heuristics*, **11**(December), 393–419.
- Beyer, H.-G. 2001. *The theory of evolution strategies*. Natural Computing Series. Springer-Verlag.
- Bramel, J., & Simchi-Levi, D. 1997. On the effectiveness of set covering formulations for the vehicle routing problem with time windows. *Operations research*, **45**, 295–301.
- Campbell, A. M., Clarke, L. W., & Savelsbergh, M. W. P. 2001. *Inventory routing in practice*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics. Pages 309–330.
- Chan, L. M. A., Federgruen, A., & D. Simchi-Levi. 1998. Probabilistic analyses and practical algorithms for inventory-routing models. *Operations research*, **46**(1), 96–106.
- Coello Coello, C. A. 1999. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and information systems*, **1**, 269–308.

- Coello Coello, C. A. 2006. Evolutionary multi-objective optimization: a historical view of the field. *Computational intelligence magazine, iee*, **1**(1), 28 – 36.
- Concorde Home Page. *Concorde TSP solver*. <http://www.tsp.gatech.edu/concorde>. Acessado em 01/07/2010.
- Deb, K. 2001. *Multi-objective optimization using evolutionary algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Ieee transactions on evolutionary computation*, **6**(2), 182–197.
- Deb, K., Mohan, M., & Mishra, S. 2003. Towards a quick computation of well-spread Pareto-optimal solutions. *Evolutionary multi-criterion optimization*, 68.
- Edgeworth, F. Y. 1881. *Mathematical physics*. P. Keagan, London, England.
- Fogel, L. J. 1999. *Intelligence through simulated evolution: forty years of evolutionary programming*. New York, NY, USA: John Wiley & Sons, Inc.
- Fonseca, C. M., & Fleming, P. J. 1993. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. *Pages 416–423 of: Proceedings of the 5th international conference on genetic algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Haimes, Y. Y., Lasdon, U., & Wismer, D. A. 1971. On a bicriterion formulation of the problems of integrated system identification and system optimization. *Ieee trans syst man cybern*, **1**(3), 296–297.
- Horn, J., Nafpliotis, N., & Goldberg, D. E. 1994. A niched Pareto genetic algorithm for multiobjective optimization. *Evolutionary computation*, **1**, 82–87.
- Moscato, P. 1989. *On evolution, search, optimization, genetic algorithms and martial arts - towards memetic algorithms*. Tech. rept. C3P 826 (Caltech Concurrent Computation Program), Pasadena: California Institute of Technology.
- Pareto, V. 1896. *Cours d'economie politique, volume i and ii*. F. Rouge, Lausanne.
- Romero, C. 2004. A general structure of achievement function for a goal programming model. *European journal of operational research*, **153**(3), 675–686.
- Srinivas, N., & Deb, K. 1994. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, **2**, 221–248.

- Zitzler, E., & Thiele, L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *Ieee trans. evolutionary computation*, **3**(4), 257–271.
- Zitzler, E., Laumanns, M., & Thiele, L. 2001. *SPEA2: Improving the strength Pareto evolutionary algorithm*. Tech. rept.